bajtek

Z MIKROKOMPUTEREM NA TY

NR 2 Cena 60 zł
PAŹDZIERNIK – 85



PIERWSZE KROKI

WITAJ „BAJTKU", PUNKTY DLA „BAJTKA", LUDZIE CHCĄ W TO GRAĆ – to tylko niektóre z tytułów, jakimi pojawienie się naszego pisma powitali koledzy-dziennikarze z gazet i tygodników. Dziękujemy za te wstępne słowa otuchy!

Otuchą w stawianiu pierwszych kroków, ale i zobowiązaniem do podnoszenia poziomu pisma jest dla nas ponad 1,5 tys listów, jakie nadeszły do redakcji „Sztandaru Młodych" po ukazaniu się pierwszego numeru BAJTKA. Dziękujemy za nie serdecznie i zapowiadamy jednocześnie, iż od przyszłego numeru rozpoczniemy drukowanie w BAJTKU specjalnych kolumn listów Czytelników i odpowiedzi redakcji. Rubrykę tę prowadził będzie Marcin Waligórski, student drugiego roku informatyki Uniwersytetu Warszawskiego (tel. 21-98-34).

W wielu listach i telefonach pytano nas o szanse zdobycia kolejnych numerów BAJTKA, w tym zwłaszcza o możliwość prenumeraty. Odpowiadam: czynimy obecnie staronia, aby od stycznia 1986 roku BAJTEK ukazywał się w większym nakładzie. Gdy tylko zostaną te sprawy ustalone – będzie można przyjmować prenumeratę, o czym, oczywiście, nie omieszkamy poinformować wcześniej Czytelników. Cierpliwości!

Cały czas oczekujemy na kolejne propozycje współpracy. Chcemy, żeby BAJTEK był w dużej mierze redagowany przez samych Czytelników. Nie stawiamy żadnych wymogów typu statusu akademickiego, czy limitu wieku. Uważamy, że ciekawe pomysły może mieć zarówno 14-latek, jak członek Akademii Nauk. Ważna jest oryginalność myślenia i chęć przekazania swych doświadczeń innym.

Często spotykamy się z pytaniem: czym chce być BAJTEK? Do jakiej grupy odbiorców chce trafiać? Powtórzmy więc deklarację z pierwszego numeru: ambicją zespołu redagującego BAJTKA jest – najogólniej mówiąc – zwalczanie analfabetyzmu mikrokomputerowego w Polsce. Chcemy więc, co jest naturalne, trafiac do 12–20-latków, ale także do wszystkich, którzy postanowili przełamać w sobie lęk przed komputerem osobistym. Oczywiście – znajdą u nas coś ciekawego dla siebie również osoby bardziej zaawansowane w programowaniu i obsłudze komputerów. Ale przede wszystkim będziemy uczyć elementarza, przynajmniej na razie.

Przyzwoity elementarz powinien być wydany na dobrym papierze, przyciągać estetyką i pełną gamą kolorów. Wszystko wskazuje na to, że takim właśnie już od stycznia 1986 roku będzie BAJTEK. Szczegóły – za miesiąc.

WALDEMAR SIWIŃSKI



BAJTEK — dodatek specjalny „Sztandaru Młodych" i „Odrodzenia". Wydawca: Krajowe Wydawnictwo Czasopism RSW „Prasa-Książka-Ruch", ul. Noakowskiego 14, 00-666 Warszawa, tel. 25-72-94. Cena 60 zł. Zam. 2266. N-16



Ze STEVE WOZNIAKIEM — współzałożycielem firmy APPLE rozmawiają GREGG WILLIAMS i ROB MOORE z redakcji miesięcznika BYTE (czyt. bajt) — najbardziej znanego pisma mikrokomputerowego świata. Wywiad ukazał się w numerach z grudnia '84 i stycznia '85 roku.

HISTORIA ZŁOTEGO JABŁUSZKA

● BYTE: Jak słyszeliśmy, mikrokomputer Apple I skonstruowałeś w 1975 r., będąc pracownikiem firmy Hewlett-Packard. Jak do tego doszło?

STEVE WOZNIAK: Komputerami i elektroniką interesowałem się od dzieciństwa. W szkole średniej studiowałem obwody telewizorów i na papierze zaprojektowałem ponad 50 komputerów, choć oczywiście nie było mnie stać na kupienie części potrzebnych do ich zbudowania. W 1970 roku, kiedy to większość ludzi nie mogła sobie pozwolić na zakup monitora, zaprojektowałem układ pozwalający podłączyć komputer do oscyloskopu, tak by na jego ekranie ukazywały się litery. Zajmowałem się także oprogramowaniem — chciałem wiedzieć, jak pisze się kompilatory takich języków jak FORTRAN i BASIC, analizowałem je i robiłem stosy notatek. Byłem papierowym samoukiem, który nigdy nie miał szansy sprawdzić swej wiedzy w praktyce.

Również lata college'u poświęciłem na pogłębianie swej wiedzy komputerowej (tylko nieliczne college rozpoczynały wówczas oferowanie studentom tego typu programu nauczania). Swój trzeci rok studiów spędziłem na University of California w Berkeley.

Sytuacja materialna zmusiła mnie do rocznej przerwy w nauce dla zgromadzenia pieniędzy potrzebnych do opłacenia dalszych studiów. Zamierzałem podjąć pracę technika, miałem jednak szczęście: zostałem przyjęty przez Hewlett-Packarda i pracując w tej firmie uzyskałem uprawnienia inżynierskie. Tak więc moja kariera zawodowa uzyskała nieoczekiwane przyspieszenie — zacząłem zdobywać doświadczenia w prawdziwym projektowaniu, poznałem tajniki planowania układów scalonych i ich działania. Z czasem zainteresowania moje zaczęły się zmieniać i nim się zorientowałem, odszedłem zbyt daleko, by móc wrócić do Berkeley. Próbowałem kontynuować studia w stanowym uniwersytecie w San Jose, ale okazało się, że praca pozostawia mi na to zbyt mało czasu, zwłaszcza że program nauczania nie w pełni był zgodny z kierunkiem moich poprzednich studiów, tak więc w praktyce uzyskanie stopnia zajęłoby mi dalsze cztery lata. W rezultacie formalnie nigdy nie ukończyłem studiów.

W istocie nadal pozostawałem elektronikiem — hobbystą. W ciągu następnych trzech lat przestałem się nawet interesować minikomputerami, gdyż całymi dniami robiłem układy kalkulatorowe u Hewlett-Packarda, a nocami chałturzyłem robiąc projekty dla innych firm.

Po raz pierwszy grę komputerową — Pong ujrzałem w pobliskiej kręgielni. Poszedłem do domu i zaprojektowałem własną jej wersję. Potem zaprojektowałem inną grę. Nazwałem ją Breakout. Stała się później przebojem Atari. Tak więc w elektronikę, jako hobby byłem zaangażowany praktycznie nieustannie, a u Hewlett-Packarda zdobyłem całkiem spore doświadczenie w projektowaniu układów scalonych.

Mniej więcej w tym samym czasie powstał Homebrew Computer Club i przypadkowo zostałem zaproszony na jego pierwsze spotkanie. Odkryłem tam tłum dzieciaków z okolicznych szkół średnich, wiedzących absolutnie wszystko na temat mikroprocesorów i języków maszynowych. Mieli oni te same problemy z jakimi ja

borykałem się trzy lata wcześniej. Odkryłem więc nagle, że mikrokomputery są w istocie tym samym, co minikomputery i natychmiast zacząłem je rozumieć.

● **Wspomniałeś, że zaprojektowałeś dla Atari grę w Breakout. Jak to się stało?**

— W tym czasie Steve Jobs pracował dla Atari. Nolan Bushnell ciągle wściekał się na niego, gdyż wszystkie ich nowe gry wymagały od 150 do 170 układów scalonych, natomiast firmie zależało na obniżce kosztów. Wówczas ujrzał moją wersję gry w Pong, wymagającą jedynie 30 układów. Kupił ją i zapowiedział, że jeśli uda nam się stworzyć wersję gry w Breakout wymagającą mniej niż 50 układów, płaci 700 dolarów, a jeśli uda nam się zejść poniżej 40 układów — dostaniemy 1000 zielonych.

Atari nie postawiło nam żadnego terminu na wykonanie tej pracy, narzucił go jednak Steve: musiałem tę robotę wykonać w cztery dni, gdyż Steve musiał złapać samolot do Oregonu. Byłem w naszym zespole projektantem i inżynierem, a Steve był montażowcem i kontrolerem.

Zdążyliśmy, dostaliśmy działającą płytkę. Mój pierwszy projekt wymagał 42 „kości", ale ostatecznie okazało się, że potrzeba ich 44. Byliśmy już zbyt zmęczeni, by szukać sposobów dalszego zmniejszenia ich liczby. Musieliśmy się więc zadowolić siedmiuset zielonymi.

APPLE I

● **Jak doszło do powstania Apple I?**

— Pracowałem wówczas nad oprogramowaniem używając terminala współpracującego z lokalną siecią komputerową. Czasami czułem potrzebę dokończenia pracy w domu. Zdecydowałem się w końcu własnoręcznie zaprojektować i zbudować domowy terminal TV wraz z modemem, co pozwoliło mi łączyć się z komputerem i grać w gry. Byłem więc prawdziwym piratem. Nauczyłem się włamywać do sieci ARPA i poprzez nią korzystać z komputerów z całego kontynentu. Gdy pojawiło się pierwsze pismo mikrofanów BYTE, czekałem przed kioskiem, by wydać dolara na pierwszy numer.

W naszym laboratorium u Hewlett-Packarda mieliśmy biurkowy komputer typu 9380 z wbudowanym BASIC-em. Został on zaprojektowany z myślą o naukowcach i kosztował ponad 10.000 dol. Nie można go więc było nazwać komputerem osobistym, ale pozwalał używać BASIC-u natychmiast po włączeniu do kontaktu. To stało się też moim celem przy Apple I — aby można było usiąść, włączyć go i przystąpić do pracy. To właśnie w Apple I było najważniejsze.

Jego charakterystyka uzależniona była głównie od możliwości terminala TV. W tym czasie powszechnie używanym urządzeniem wejścia—wyjścia był dalekopis typu ASR-33. Był on standardem przez ponad 10 lat i firmy minikomputerowe dopiero zaczynały posługiwać się wideoterminalami.

W 1975 roku wideoterminale projektowano z rejestrami przesuwnymi, gdyż pamięci RAM były zbyt drogie. Trzeba więc było instalować cały zestaw takich rejestrów i przesuwać je, stale wysyłając znaki na ekran. W rezultacie Apple I pracował stosunkowo powoli. Mógł on wysłać na ekran jedyne 60 znaków na sekundę: po jednym znaku podczas każdego cyklu wyświetlanie ekranu. Moim celem było jednak głównie oszczędzanie na częściach, a nie zwiększanie możliwości maszyny.

● **Czy można więc nazwać Apple I komputerem z prawdziwego zdarzenia?**

— Tak, choć jego właściwości różniły się nieco od możliwości Apple II i innych późniejszych komputerów osobistych: był on powolny, czysto tekstowy, ale i tak o wiele szybszy od dalekopisów, do jakich byliśmy przyzwyczajeni — pisały one jedynie 10 znaków na sekundę. Terminale telewizyjne w tych dniach dopiero zaczynały zdobywać popularność.

● **Przede wszystkim były one niezwykle drogie...**

— Tak, a ja musiałem budować tanio, gdyż nie miałem pieniędzy. Używałem najstarszych, najtańszych, przecenionych części, jakie tylko można było zdobyć. W piśmie Radio-Electronics ukazał się w tym czasie artykuł zatytułowany TVT1 i było to wówczas wielkie wydarzenie dla hobbystów. Don Lancaster przedstawił w nim skomplikowaną konstrukcję, wykorzystującą całe tony części i bramek logicznych, ale moim zdaniem był to zły projekt. Ja miałem już wprawę w tworzeniu dowcipnych rozwiązań, oszczędzających części, w czym bardzo przydatna była wprawa w wykorzystywaniu zależności czasowych pracy monitora TV. Z czasów szkolnych pamiętałem, że odbiorniki TV są projektowane z dużą rezerwą, tak więc nawet jeśli moje układy nie mieściły się w teoretycznie dostępnym czasie, to mimo to dobrze współpracowały z większością telewizorów i monitorów.

● **Nie troszczyłeś się więc zbytnio o precyzyjny rytm pracy twoich urządzeń.**

— Prawdę mówiąc nie — byłem konsekwentnym hobbystą-praktykiem. Nie tworzyłem w końcu wyrobu do masowej produkcji, lecz coś, co miało działać u mnie w domu z moim własnym telewizorem. Komputer używał więc procesora 6502 połączonego z terminalem poprzez układ zwany PIA, który mógł także czytać klawiaturę. Kupiłem więc z ogłoszenia w gazecie przecenioną klawiaturę za 60 dol. Emulowała ona dalekopis i robiła wszystko, czego oczekiwałem, tak więc podłączyłem ją do mej maszyny.

Moim głównym problemem była pamięć. W tym czasie z przeceny można było kupić jedynie „kości" 2102 z 1 kB statycznej RAM. Mój komputer był jednak już zaprojektowany i miałem już napisany BASIC na 4 kB, tak więc pożyczyłem od przyjaciela płytkę z 4 kB statycznej RAM, zbudowanej z układów 2102 i mogłem zacząć testować swoje pomysły. Okazało się, że mój BASIC spisuje się całkiem nieźle, nadal jednak chciałem zdobyć pamięci dynamiczne, ponieważ umożliwiałoby mi to poważne zmniejszenie liczby wykorzystywanych układów.

Steve był zainteresowany moimi pomysłami i któregoś dnia zapytał mnie po prostu: „Dlaczego, u licha, nie używasz tych nowych 16-nóżkowych pamięci dynamicznych?" Widziałem już je co prawda u Hewlett-Packarda, ale była to wówczas absolutna nowość, a mnie nie było stać na żadne części poza takimi, jakie udawało mi

DOKOŃCZENIE NA STR. 26

KILKA UWAG O ALGORYTMACH

Algorytmem nazywa się przepis postępowania, prowadzący w sposób automatyczny do rozwiązania określonego problemu. Przepis ten powinien być na tyle precyzyjny, aby posługiwanie się nim polegało wyłącznie na wykonywaniu jego instrukcji. Zakłada się przy tym, że pewne instrukcje pierwotne tego przepisu są wykonalne tzn., że są uprzednio zdefiniowane.

Każdy algorytm można opisać w sposób słowny używając poleceń typu: wykonaj...; jeśli jest prawdą..., to wykonaj...; zakończ; zacznij od początku; itd. Najłatwiejsze i najbardziej zrozumiałe jest przedstawienie algorytmu w postaci graficznej (schematów blokowych), zgodnej z ogólnie przyjętą konwencją. I tak opis instrukcji (wykonaj...) umieszcza się zazwyczaj w prostokącie rys. 1a, testy (jeśli jest prawdą..., to wykonaj...) w rombach lub sześciokątach rys. 1b, a oznaczenie startu i stopu algorytmu w okręgach rys. 1c. Niekiedy wyróżnia się także polecenia wprowadzania informacji z zewnątrz (np. z klawiatury lub dysku) i wyprowadzania jej (np. na monitor lub drukarkę). Umieszcza się je wówczas w elementach przedstawionych na rys. 1d.

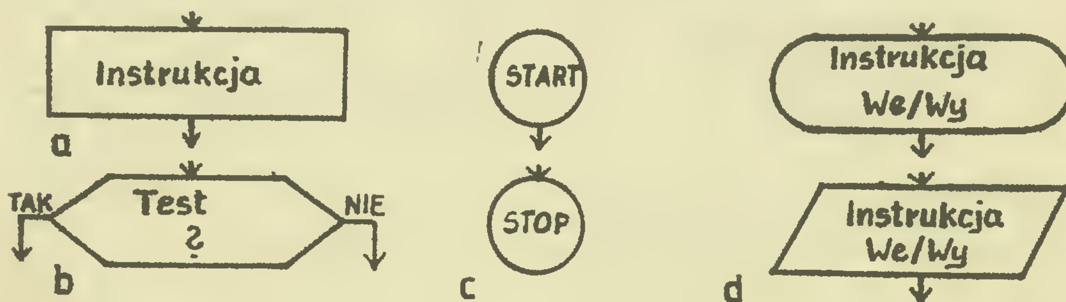
Poszczególne elementy opisu graficznego łączy się ze sobą strzałkami, które określają kolejność operacji wykonywanych w algorytmie. Strzałki wychodzące z pól opisu testu są etykietowane słowami „tak" lub „nie" i wskazują na następne czynności w zależności od wyniku testu.

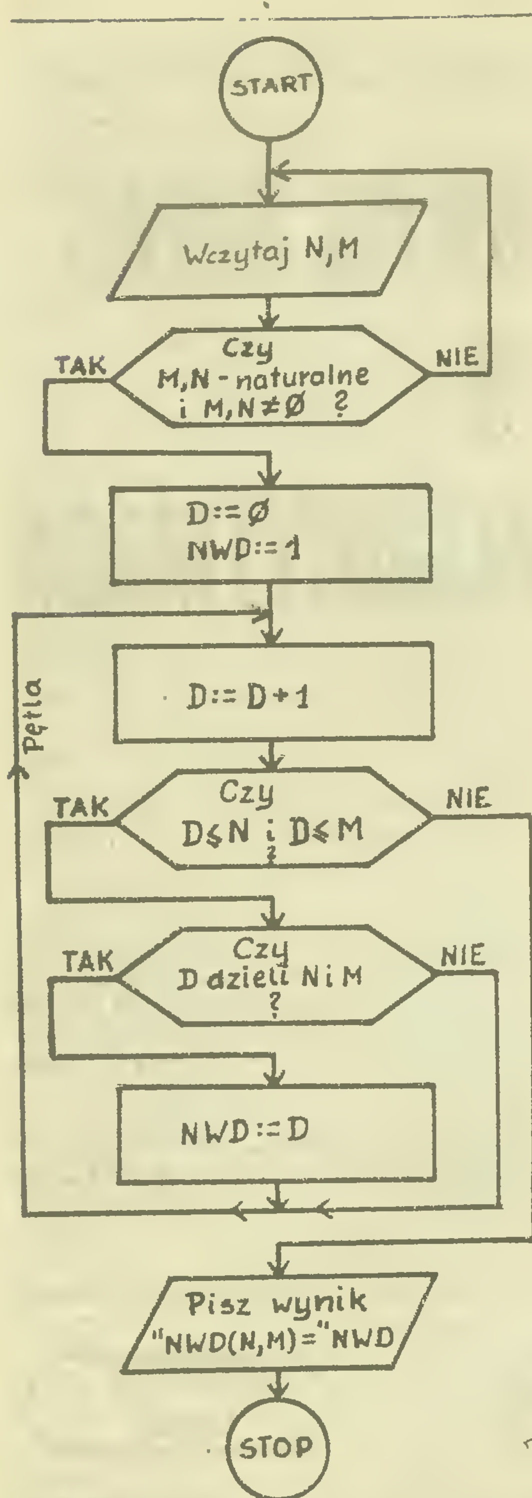
Instrukcje lub testy mogą być opisywane słownie lub za pomocą wyrażeń i symboli. Im bardziej opis algorytmu jest sformalizowany, tzn. zapisany w konwencji pewnego języka programowania, tym łatwiejsze jest przejście od algorytmu do programu. Przekształcanie algorytmu w program (w określonym języku programowania) to często bardzo pracochłonne zajęcie, lecz jest wyłącznie czynnością wtórną. Naprawdę twórcze jest napisanie prawidłowego algorytmu. Nieco upraszczając, możemy przyjąć, że prawidłowy algorytm to taki, który gwarantuje osiągnięcie postawionego celu zawsze w skończonej liczbie kroków.

Każdy problem można rozwiązać na kilka różnych sposobów. Istnieje więc wiele algorytmów. Jak spośród nich wybrać najlepszy, optymalny? Jeśli założymy, że najlepszy to ten, który doprowadza do celu po wykonaniu najmniejszej ilości operacji, to wybór staje się jednoznaczny.

Spróbujmy rozwiązać następujące zadanie zaczerpnięte z teorii liczb: dane są dwie liczby

Rys. 1. Elementy składowe schematu blokowego





Rys. 2. Schemat blokowy algorytmu „siłowego" znajdowania NWD dwóch liczb naturalnych

naturalne N i M, należy znaleźć ich największy wspólny dzielnik (NWD) tzn. największą liczbę naturalną, która dzieli N i M bez reszty. Dla przykładu NWD dla liczb 15 i 10 jest równy 5. Rozwiązaniem, które natychmiast rzuca się w oczy, jest branie kolejnych liczb od jednego do min (N, M*) sprawdzanie czy dzielą N i M bez reszty a następnie wybranie największej spełniającej ten warunek (algorytm na rys. 2). W ten sposób pętla zostanie wykonana dokładnie min (N, M) razy.

Zachodzi pytanie czy można wykonując mniej operacji osiągnąć ten sam rezultat? Okazuje się, że można, zrobił to kiedyś Euklides. Jedno z jego rozwiązań ma następujące matematyczne uzasadnienie:

Przyjmijmy, że N jest większe lub równe M. Wtedy N można przedstawić jako N = q × M + R (gdzie q nazywa się ilorazem, a R resztą z dzielenia N przez M) — oczywiście R<M. Jeśli R = 0, największym wspólnym dzielnikiem jest liczba M, jeśli jednak R jest większe od 0 to wówczas każdy dzielnik liczby N i M jest jednocześnie dzielnikiem R (bo R = N — q × M, a wspólny dzielnik zawsze możemy wyciągnąć przed nawias).

W szczególności dotyczy to największego wspólnego dzielnika. A więc

NWD (N, M) = NWD (M, R)

W tej sytuacji przyjmijmy następujący sposób postępowania.

Oblicz R — resztę z dzielenia N przez M

Jeśli R = 0 to NWD (N, M) = M i koniec algorytmu.

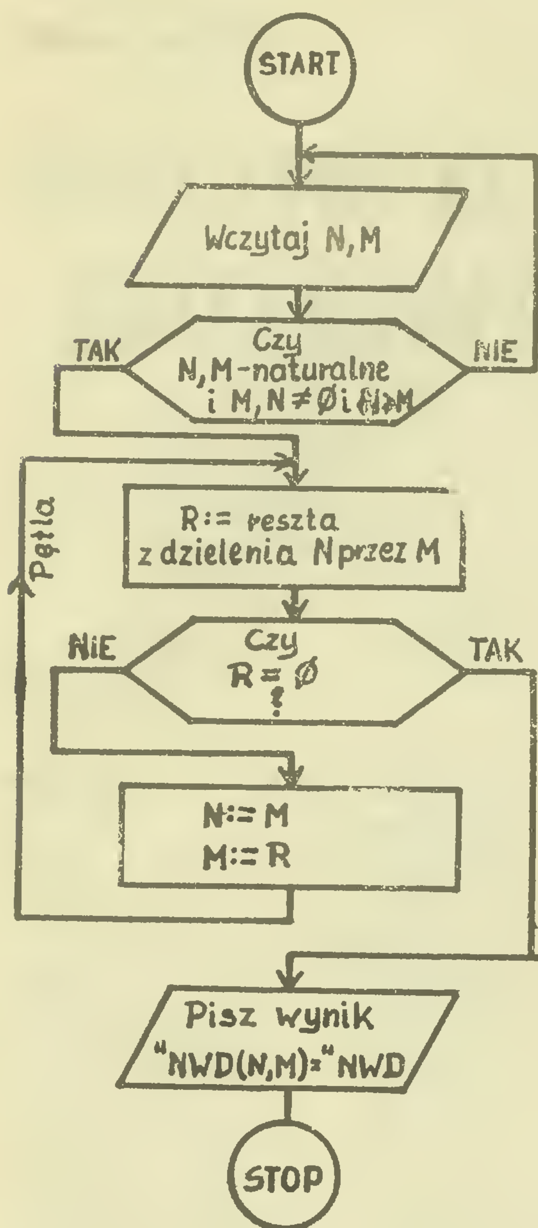
Jeśli R ≠ 0, to podstaw M w miejsce N i R w miejsce M i powtórzyć postępowanie od początku.

Schemat blokowy algorytmu Euklidesa został przedstawiony na rys. 3. Zostało udowodnione, że pętla zostanie wykonana maksymalnie 1.44041 log₂ (min (N, M)) razy, czyli o wiele mniej w porównaniu z poprzednim „siłowym" algorytmem.

Prześledźmy działanie algorytmu dla N = 825 i M = 420

N	M	R
825	420	405
420	405	15
405	15	0

Pętla zostanie wykonana tylko trzy razy!



Rys. 3. Schemat blokowy algorytmu Euklidesa.

Wiadomo, że istnieje* lepsze rozwiązanie, o maksymalnej liczbie iteracji równej $\log_2$ (min. (N,M). Nie da się jednak udowodnić, że dany algorytm jest optymalny. Może ktoś z czytelników pokusi się o wymyślenie jeszcze lepszego algorytmu.

WOJCIECH PENCZEK

*) min. (N, M) oznacza mniejszą z liczb N i M.

Czytelnicy piszą

„Mam 19 lat i maturę w kieszeni. Uczę się w studium policealnym. Jestem naprawdę szczęśliwy, że ktoś zaczął wreszcie wydawać pismo o tematyce komputerowej. Powinno się ono pojawić już ze dwa lata temu. Dobrze, że jest!"

Artur Telka z Kielc

„Mam 14 lat. Udało mi się kupić pierwszy numer BAJTKA. Jest bardzo ciekawie redagowany i pisany językiem zrozumiałym nawet dla takiego nowicjusza jak ja. W mojej okolicy i województwie nie ma niestety klubów komputerowych, a rodziców nie stać na wyłożenie 100 tys. zł na moje „zachcianki", czyli mikrokomputer. Gdzie mógłbym rozwijać swoje zainteresowania?"

Krzysztof Łyp z Łukowa

„W mojej szkole ruszyło kółko komputerowe. Uczęszczają do niego chłopcy i jedna dziewczyna — czyli ja. Niestety koledzy szybciej orientują się w tych zagadnieniach. Liczę na BAJTKA! Nie udało mi się do tej pory go kupić, a w rękach miałam go dosłownie 3 minuty. BAJTEK był pożyczany przez kolegę od kolegi kolegi. Ale nawet w ciągu tych 3 minut szalenie mnie zainteresował."

Katarzyna Ziemnicka (lat 16) z Mińska Mazowieckiego

„Wspaniała inicjatywa, BAJTEK, jak magiczne zaklęcie, pozwala rozpoznać entuzjastów zafascynowanych nową techniką. Łączę wyrazy uznania."

Igor Glinieckі z Lęborka

„Z cyklem artykułów z dziedziny mikroinformatyki „Sztandar Młodych" stał się dla mnie i wielu innych hobbystów jednym z poczytniejszych czasopism. A już zupełną „bombą" jest pojawienie się BAJTKA."

Jacek Wypychowicz z Kielc

Wszystkich zainteresowanych podnoszeniem kultury informatycznej w naszym kraju, osoby posiadające doświadczenie w użytkowaniu mikrokomputerów, wszystkich, którzy chcą z BAJTKIEM współpracować lub choćby wymienić uwagi, prosimy o kontakt z redakcją. Adres do korespondencji:

Redakcja „Sztandaru Młodych"
00-687 Warszawa
ul. Wspólna 61
BAJTEK

Telefony:
29-51-06 (Waldemar Siwiński),
28-52-71 wew. 259
(Roman Poznański).

I love sinclair Computer

JAK ZX SPECTRUM CZYTA KLAWIATURĘ

Klawiatura jest jednym z najważniejszych urządzeń We Wy (I O) każdego komputera. W przypadku ZX Spectrum, jest ona podzielona na 8 rzędów po 5 klawiszy.

Każdy z tych rzędów można osobno zaadresować. Pozwala to na sprawdzenie czy którys z pięciu klawiszy został wciśnięty, np. wybranie szóstego wiersza (zaczynamy liczyć od 0), daje możliwość sprawdzenia H,J,K,L i ENTER. Osiem wierszy klawiatury podłączono do szyny adresowej od linii A8 do linii A15, a więc do starszego bajtu adresu. Adres danego wiersza generowany jest zgodnie z wzorem

ADRES = 254 + 256 × $(255-2^n)$

gdzie wykładnik potęgi n jest numerem wiersza (od 0 do 7).

Kolumny matrycy podłączone są do szyny danych mikrokomputera przez układ ULA (ang. Uncommited Logic Array), np. klawisze: 5,T,G,6,Y,V,H,B są podłączone do linii D4, a klawisze od 1 do SPACE — do linii D0. Naciśnięcie dowolnego klawisza z danej kolumny powoduje, że na linii danych tej kolumny pojawia się niski stan napięcia — odpowiadający logicznemu zeru. Pojawia się on tylko wtedy, gdy wiersz, w którym znajduje się klawisz jest zaadresowany. Tak więc adres oraz odczytana informacja pozwalają na jednoznaczną interpretację, który klawisz został wciśnięty.

Komputer przeszukuje klawiaturę 50 razy na sekundę. Robi się to w następujący sposób: generuje kolejno adresy wierszy klawiatury od pierwszego wiersza (adres 32766) do ostatniego (adres 65278) i sprawdza informacje na szynie danych. Wykrycie zera na jednym z bitów od D0 do D4, oznacza wciśnięcie określonego klawisza.

★

Sprawdzenie klawiatury odbywa się niezależnie od wykonywania innego programu np. w BASIC-u. Stąd wniosek, że podczas wykonywania programu w BASIC-u mikrokomputer niepotrzebnie traci czas na przeszukiwanie klawiatury. Przeszukiwanie klawiatury można wyłączyć wykonując instrukcję maszynową Z80-DI (ang. disable interrupts), a włączyć instrukcją EI (ang. enable interrupts). W praktyce robi się to w następujący sposób:

```
11 CLEAR adres — 1
1Ø GOSUB 99ØØ
. .
. . .
. .
989Ø GOTO 992Ø
99ØØ RANDOMIZE USR adres: REM wyłącz przerwania
99ØØ RETURN
992Ø RANDOMIZE USR adres + 2: REM włącz przerwania.
```

Pod adresem „adres" należy umieścić następujący program w języku wewnętrznym, np. instrukcją POKE

adres	DI	POKE adres, 243
adres + 1	RET	POKE adres + 1, 201
adres + 2	EI	POKE adres + 2, 251
adres + 3	RET	POKE adres + 3, 201

Linia 1 programu rezerwuje miejsce na program w asemblerze. Następnie wyłącza się odczytywanie klawiatury, a po wykonaniu programu w BASIC-u włącza się ponownie.

★

Rozważmy następujący przykład: Wciśnięty zostaje klawisz K. Znajduje się on w szóstym wierszu. Jeżeli zostanie wygenerowany adres

254+256 $(255-2^6)$ =49150

jest to równoznaczne z zaadresowaniem wiersza, w którym znajdują się znaki od H do ENTER. Jeżeli żaden klawisz nie byłby w tym wierszu przyciśnięty, to z portu o wybranym adresie można odczytać wartość 191, a nie jakby można było oczekiwać 255. Wynika to z faktu, że bit D6 szyny danych wykorzystuje się do odczytu stanu gniazda EAR, współpracującego

ULA

SZYNA ADRESOWA

	DØ	D1	D2	D3	D4	
A 15	SPA CE	SYM SHIFT	M	N	B	IN 32766
A 14	ENTER	L	K	J	H	IN 49150
A 13	P	O	I	U	Y	IN 57342
A 12	Ø	9	8	7	6	IN 61438
A 11	1	2	3	4	5	IN 63486
A 10	Q	W	E	R	T	IN 64510
A 9	A	S	D	F	G	IN 65022
A 8	CAPS SHIFT	Z	X	C	Y	IN 65278

z magnetofonem. Jeżeli magnetofon nie jest podłączony, to przez cały czas ma on wartość logicznego zera (niski stan napięcia), wszystkie pozostałe bity mają stan wysoki — logiczna jedynka. Odczytana wartość jest więc równa $191=255-2^6$. Zilustrować to można w...nując poniższy program

```
1Ø PRINT IN 4915Ø: PAUSE 1ØØ
2Ø GOTO 1Ø
```

W efekcie jego wykonania na ekranie będzie wyświetlona zawartość portu 49150. W omawianym przypadku wciśnięty jest klawisz K. Odczytana przez IN 49150 wartość będzie wynosiła 187. Wartości te zmieniają się zgodnie ze wzorem

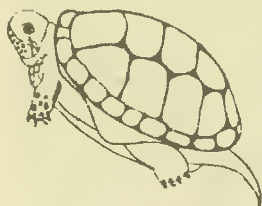
WARTOŚĆ $= 191 - 2^K$ jest elementem (0, 1, 2, 3, 4)

Wykładnik potęgi K odpowiada numerowi kolumny podłączonej przez ULA do odpowiedniego miejsca na szynie danych. Jeżeli zostaje wciśnięty inny klawisz z tego wiersza p. H, to na ekranie zostanie wyświetlona liczba $175 = 191 - 2^4$. Aby przekonać się o słuszności powyższych wywodów należy wykonać podany wyżej program kilkakrotnie, zmieniając adresy w instrukcji IN.

Klawiatura ZX Spectrum może pracować w roznych trybach (LIC, E, G i K). Informacje o tym, w którym trybie aktualnie znajduje się klawiatura, komputer przechowuje w komórce pamięci o adresie 23617. Adres ten jest przypisany do zmiennej systemowej o nazwie MODE. Zmienna ta przyjmuje wartości 0 dla trybu LIC, 1 dla trybu E, 2 dla trybu G oraz 4 dla trybu K. System operacyjny, znajdujący się w pamięci stałej ROM, na podstawie stanu tej komórki pamięci, informacji o tym, który klawisz został wciśnięty oraz informacji o tym czy był wcześniej wciśnięty klawisz SYMBOL SHIFT lub CAPS SHIFT, ustala ostatecznie jaka funkcja zostaje wybrana.

Są to podstawowe wiadomości dotyczące czytania klawiatury przez ZX Spectrum. Kluczem do nich jest zamieszczony rysunek. Bardziej szczegółowe informacje można znaleźć np w książce „Spectrum Hardware Manual"

KRZYSZTOF KURYŁOWICZ
DARIUSZ MADEJ



LOGO LOGO LOGO

Kontynuujemy suchy opis języka LOGO, szersze komentarze odkładając do dalszych numerów. Podawany tu opis dotyczy wersji Sinclair Logo SOLI/LCSI na Spectrum, różnice pomiędzy wersjami są jednak niewielkie i dotyczą kwestii drugorzędnych, np. sposobu korzystania z edytora lub współpracy z drukarką.

Liczby

Liczby w LOGO są również słowami, nie muszą być jednak poprzedzane znakiem cudzysłowu: 45 LOGO rozumie jako liczbę 45, a nie jako wywołanie procedury o nazwie "45. Również nazwy wartości logicznych: TRUE lub FALSE rozumiane są zawsze jako wyrażenia logiczne, tak więc są to jakby dwie podstawowe procedury o takim mniej więcej brzmieniu:

TRUE
FALSE

```
TO TRUE
OUTPUT "TRUE
END
```

(dla nie znających angielskiego: TRUE znaczy prawda, FALSE — fałsz). Wartości logiczne uzyskujemy w LOGO m.in. korzystając z procedur, będących pytaniami pod adresem translatora, których nazwy kończą się literą P, np. DEFINEDP "Drzewo daje w odpowiedzi TRUE lub FALSE. Wartości logiczne przyjmują również wyrażenia arytmetyczne ze znakiem nierówności lub równości. Wartości te mogą być poprzez OUTPUT przekazywane do procedur nadrzędnych i być przez nie wykorzystywane do podejmowania decyzji o wyborze drogi dalszej realizacji programu.

Liczby w LOGO można oczywiście dodawać, odejmować, mnożyć i dzielić. Dodawanie można wykonać na dwa sposoby: wykorzystując operator zewnętrzny SUM lub wewnętrzny +. Tak więc wyrażenie:

SUM

```
SUM 5 2
```

jest równoważne 5 + 2, a wyrażenie

```
(SUM 5 2 8 12)
```

jest równoważne 5 + 2 + 8 + 12,
podobnie funkcjonują operatory PRODUCT (mnożenie) i DIV (dzielenie), przy czym dzielić można oczywiście tylko dwie liczby!

PRODUCT
DIV

Wynik dzielenia jest dokładny, z dokładnością, z jaką komputer wykonuje obliczenia arytmetyczne (w Spectrum ok. 9 cyfr po przecinku).

INT

Jeśli interesuje nas część całkowita wyniku dzielenia trzeba skorzystać z procedury INT, natomiast resztę z dzielenia można otrzymać bezpośrednio przy pomocy procedury REMAINDER, np:

REMAINDER

```
PR REMAINDER 25 4
```

daje w odpowiedzi 1.

ROUND

+ −

* /

COS
COSINE
SIN
SINE
TAN
COT
ARCTAN
ARCCOS
ARCSIN
ARCCOT

SQRT
RANDOM

Innym sposobem zamiany liczby rzeczywistej na całkowitą jest operacja ROUND, której wynikiem jest najbliższa liczba całkowita, a więc zaokrąglenie.

Operatorowi SUM równoważny jest operator + (lub —), umieszczany (wyjątek od standardowej składni LOGO!) między argumentami, podobnie jak operatory * (razy) i / (dzielenie).

Wartość wyrażenia arytmetycznego zawsze obliczana jest przed pobraniem parametru dla procedury poprzedzającej wyrażenie, a w ramach wyrażenia pierwszeństwo mają operatory zewnętrzne, a następnie mnożenie i dzielenie.

Zestaw operacji matematycznych uzupełniają funkcje trygonometryczne: COS lub COSINE, SIN lub SINE, TAN lub TANGENT, COT lub COTANGENT oraz ich odwrotności: ARCCOS, ARCSIN, ARCTAN i ARCCOT. Argumentem dla funkcji trygonometrycznych są wartości kątów wyrażone w stopniach!

Istnieje także możliwość obliczenia pierwiastka kwadratowego SQRT oraz posłużenia się generatorem liczb losowych RANDOM, który zwraca z jednakowym prawdopodobieństwem jedną z liczb naturalnych mniejszych od argumentu lub zero:

OPERACJE NA LISTACH

ASCII
CHAR

FIRST
BUTFIRST BF
LAST
BUTLAST BL
FPUT
LPUT

ITEM

COUNT

EMPTYP
EQUALP
MEMBERP
LISTP
WORDP
NUMBERP

W poprzednim odcinku omówiliśmy podstawowe operacje, dziś kolej na bardziej złożone:

operacja ASCII podaje kod ASCII odpowiadający podanemu znakowi, a operacja CHAR odwrotnie — znak odpowiadający podanemu kodowi.

Duża grupa operacji pozwala wyodrębniać początkowe i końcowe elementy listy z myślą o kolejnym jej przetwarzaniu: FIRST wybiera pierwszy element listy lub pierwszy znak słowa, zaś BUTFIRST (lub w skrócie BF) — wszystkie poza pierwszym, analogicznie LAST — ostatni element, zaś BUTLAST (BL) — wszystkie poza ostatnim.

Podobnie operacja FPUT dołącza podany element na początek danej listy, zaś LPUT — na koniec.

Istnieje również możliwość wybrania dowolnego elementu z listy, trzeba jednak znać jego miejsce na niej: ITEM 3 [To jest nasza przykładowa lista] wybierze trzeci element listy, a więc słowo "nasza. Ustalić, na którym miejscu na liście znajduje się interesujący nas obiekt pozwala operacja COUNT.

Opracowując i przetwarzając listy możemy zadawać LOGO cały szereg pytań, np. czy dana lista jest pusta (EMPTYP), czy dwa podane obiekty są równoważne (EQUALP), czy dany obiekt jest w ogóle elementem danej listy (MEMBERP) i czy obiekt ten jest listą (LISTP), słowem (WORDP) czy też może liczbą (NUMBERP).

GRAFIKA

FORWARD FD
BACK BK
LEFT LT
RIGHT RT

SCHOWTURTLE ST
HIDETURTLE HT
SHOWNP

Żółw może się poruszać do przodu (FORWARD, FD), do tyłu (BACK, BK), w lewo (LEFT, LT) i w prawo (RIGHT, RT). Ruchy żółwia liczymy w krokach — ekran Spectrum liczy 255 na 175 kroków i w stopniach.

Żółw może być na ekranie widoczny lub nie. Ukazuje się na komendę SHOWTURTLE (lub ST), a chowa na komendę HIDETURTLE (lub HT). SHOWNP pozwala zapytać się, czy żółw jest widoczny.

Żółw poruszając się po ekranie zostawia ślad, jeśli porusza

PENUP PU
PENDOWN PD
PENERASE PE

PENREVERSE PX

się z piórem opuszczonym na papier. Pióro można podnieść (PENUP, PU) lub opuścić (PENDOWN, PD), można też zamiast pióra dać żółwiowi gumkę, by zmazywał wszystko na co natrafi (PENERASE, PE). Możliwe jest też danie mu bardziej skomplikowanego zadania — by zmazywał, gdy coś jest już narysowane lub rysował, gdy nic nie ma (PENREVERSE, PX) — czyli robił z białego czarne, a z czarnego białe.

SETPC
SETBG
SETBORDER SETBR

Żółw może rysować w ośmiu różnych kolorach, a numer koloru można mu podać przy pomocy SETPC, numer koloru tła przy pomocy SETBG, a numer koloru zewnętrznej ramki przy pomocy SETBORDER (SETBR).

PENCOLOUR PC
BACKGROUND BG

Ustalić bieżący kolor pióra można przy pomocy operacji PENCOLOUR (PC), a kolor tła przy pomocy operacji BACKGROUND (BG).

SETPOS

Żółwia można przesuwać nie tylko podając mu jego ruchy, lecz także określając pozycję, jaką ma zająć: komenda SETPOS [x y] przesuwa żółwia na miejsce o współrzędnych x y, przy czym początek układu współrzędnych znajduje się w środku ekranu!

SETHEADING SETH
TOWARDS

Komenda SETHEADING (SETH) pozwala kazać żółwiowi odwrócić się w zadanym kierunku, a komenda TOWARDS [x y] poleca mu spojrzeć w kierunku podanego punktu, a HEADING — zapytać się o kierunek od żółwia do podanego punktu.

SETX
SETY

Żółwia można też przesunąć poziomo w zadane miejsce (SETX) lub pionowo (SETY).

POSITION POS

Bieżące położenie żółwia można ustalić zadając pytanie o jego pozycję (POSITION, POS) — odpowiedź, uzyskujemy w postaci listy zawierającej dwa elementy — współrzędną x i y. Można też zapytać się tylko o jedną współrzędną: XCOR i YCOR.

XCOR
YCOR

HOME

Żółwia można też przesunąć do jego pozycji wyjściowej w środku ekranu, głową do góry, przy pomocy komendy HOME.

DOT

Niezależnie od żółwia można postawić na ekranie punkt przy pomocy komendy DOT [x y].

CLEAN

CLEARSCREEN CS

Wszystkie znajdujące się na ekranie rysunki można usunąć przy pomocy zlecenia CLEAN, które zostawia jednak żółwia na dotychczasowym miejscu. Jeśli chcemy zacząć wszystko od nowa możemy posłużyć się komendą CLEARSCREEN (CS).

WRAP
FENCE
WINDOW

Istnieją trzy możliwe sposoby reagowania komputera na opuszczenie przez żółwia ekranu. Bez uprzedniego zaznaczenia komputer po włączeniu znajduje się w trybie WRAP, w którym żółw opuszczając ekran z jednej strony ukazuje się po przeciwnej (WRAP od zawinąć). Można też ustalić tryb FENCE, w którym opuszczenie ekranu przez żółwia traktowane jest jako błąd (FENCE — płot) lub WINDOW (okno), w którym żółw może swobodnie poruszać się poza ekranem.

SETSCRUNCH

SCRUNCH

Ostatnie dwie komendy graficzne dotyczą możliwości regulowania proporcji ekranu: SETSCRUNCH [X Y] pozwala zmieniać proporcje między rozmiarami kroku w pionie i poziomie, a SCRUNCH — zapytać o ustalone proporcje. Bez osobnych ustaleń proporcje te są równe [100 100].

Na tym kończymy podstawowy opis LOGO. W dalszych numerach opiszemy jeszcze współpracę języka z urządzeniami zewnętrznymi (magnetofonem, drukarką, klawiaturą, robotem) oraz podamy przykładowe programy w LOGO

W.M.

PROGRAMOWANIE W JĘZYKU LOGIKI PROLOG cz. 1

PROLOG jest językiem nietypowym, programowanie w nim przypomina bardziej rozmowę niż operowanie nazwami zastrzeżonymi czy rozkazami jak w BASIC-u i PASCAL-u. Służy on do przetwarzania języków naturalnych, list danych itp... Szczególnie nadaje się do budowania inteligentnych baz danych o dostępie w języku naturalnym, kompilatorów, systemów strategicznych, języków problemowych czy programów sprawdzających poprawność dowodów twierdzeń.

Pełną definicję języka PROLOG opracował w 1972 r. Alain Colmeraur. PROLOG dostępny jest na wielu mikrokomputerach opartych na mikroprocesorach: Z-80 — pod systemem CP/M 80, oraz 8088/86 pod systemem MSDOS i komputerach wyposażonych w system operacyjny UNIX.

Implementacją PROLOG-u na komputer ZX-Spectrum jest micro-PROLOG. Ponieważ jego syntaktyka mogłaby się wydawać nieco zawiła dla osób nie obeznanych z gramatykami metamorficznymi, opracowano pewne rozszerzenie języka podstawowego o nazwie SIMPLE. Tak więc w pierwszej kolejności nagrywany program PROLOG a następnie SIMPLE (komendą load SIMPLE).

Rozpoczniemy od podawania przykładów w składni uproszczonej. Przykłady zaczerpnięto z książki Clark K.L., Emmals R., McCabe F.G: „A micro-PROLOG Primer" Logic Programming Associates L.T.D., 1983.

Podstawowe wyrażenia w PROLOG-u to twierdzenia i pytania. Twierdzenia określają pewien obiekt lub ich grupę oraz ustalają relacje między nimi. Oto przykład zdań poprawnych pod względem logicznym: Jan jest ojcem Piotra, Darek jest bratem Piotra. Nie są natomiast zdaniami w sensie logicznym następujące stwierdzenia: Czy jutro będzie pogoda? Chyba pójdę do kina.

Zdania: Jan jest ojcem Piotra i Darek jest bratem Piotra, moglibyśmy zapisać poprawnie pod względem syntaktycznym następująco:

&.add (Jan jest-ojcem Piotra)

&.add (Darek jest-bratem Piotra)

Słowa „jest" i „bratem" łączymy myślnikiem (lub dowolnym innych znakiem prócz spacji), aby uzyskać jedno pojęcie oznaczające relację: jest bratem.

Znak „&." ukazuje się zawsze na początku nowej linii i oznacza gotowość systemu do przyjęcia kolejnego zlecenia. Wysłanie przez system komunikatu „Error: 2" oznacza, że program SIMPLE nie został nagrany. Należy go nagrać komendą load SIMPLE i powtórzyć poprzednio wykonane operacje.

Przypuśćmy, że chcielibyśmy dodać jeszcze jedno stwierdzenie będące naturalną konsekwencją obydwu poprzednich:

&.add (Jan jest-ojcem Darka).

Otóż dla systemu PROLOG „Darek" i „Darka" są dwoma różnymi obiektami. Nieścisłość ta wynika ze specyficznej odmiany rzeczowników w języku polskim. Tak więc oddaję pod sąd Czytelników następujący dylemat:

Czy pisać przykłady w języku ojczystym, kalecząc go przy tym niemiłosiernie (za przykład niech posłużą następujące zdania w PROLOG-u: Jan ojciec Darek, Małgorzata siostra Jan, Helena ciotka Jacek...), czy też pisać je w języku, w którym wspomniany problem nie występuje (ja będę stosował język angielski). I tak nasze zdania wyglądają następująco:

&.add (John father-of Peter)
&.add (Dark brother-of Peter)
&.add (John father-of Dark)

Relacje zachodzące między obiektami można zapisywać w formie postfiksowej, tj. takiej, w której nazwa własności występuje po nazwie obiektu, posiadającego tę własność.

Henry male Henryk — płeć męska.

Jane female Jane — płeć żeńska.

O zdaniu, w którym nazwa relacji poprzedza listę obiektów, mówimy, że ma formę prefiksową.

gives (John Mary flower) — daje (Jan Marii kwiat)

reads (Mark book) — czyta (Marek książkę

Nawias zastosowano w celu oddzielenia obiektów od relacji, zaś relację, w której jej nazwa występuje między obiektami nazywamy relacją infiksową np.

Henry father-of Elizabeth Henryk ojciec Elżbiety

Podstawową formą relacji jest forma prefiksowa. Zdania dwuskładnikowe mogą również być pisane w formie prefiksowej.

Stwierdzenia:

father-of (Henry Elizabeth)
oraz
Henry father-of Elizabeth
są rownoważne.

W matematyce i logice obiekty, między którymi zachodzi relacja nazywamy argumentami. Mówimy o pierwszym, drugim itd., argumencie. Warto pamiętać, że micro-PROLOG wypisuje zdania zawsze w formie postfiksowej.

Wiedza, którą już posiadamy, pozwala nam na stworzenie małej bazy danych. Będzie ona opisywała relacje zachodzące w pewnej rodzinie. Przy okazji poznamy wszystkie komendy PROLOG-u. Nie jest ich wiele.

Wprowadźmy następujące zdania.

&.add (Elizabeth mother-of Henry)
&.add (Katherine mother-of Mary)
&.add (Ann mother-of Elizabeth2)
&.add (Ann mother-of Elizabeth 2)
&.add (Henry father-of Edward)
&.add (Jane mother-of Edward)
&.add (Henry-Snr male)
&.add (Elizabeth2 female)
&.add (Katherine female)
&.add (Mary female)
&.add (Elizabeth 2 female)
&.add (Ann female)
&.add (Female (Jane))
&.add (Male (Edward))



Ostatnie dwa zdania zapisaliśmy w formie prefiksowej. Przy pomocy zlecenia **add** możemy dodawać do naszego zbioru w każdej chwili dowolną ilość informacji w postaci zdań. Dane, które dotychczas wprowadziliśmy, są pogrupowane pod względem relacji, jakie między nimi zachodzą. PROLOG „zna" więc następujące obiekty:

Henry-Snr
Henry
Mary
Elizabeth
Elizabeth2
Ann
Edward
Jane

oraz relacje:

father-of
mother-of
male
female

Warto przy okazji nadmienić, że Elizabeth2 to Elizabeth druga, natomiast 2Elizabeth lub 2 Elizabeth to po prostu dwie Elizabeth (oczywiście te same).

Istnieje komenda, która znacznie przyspiesza wprowadzenie listy obiektów o tej samej własności. Jest nią **accept**. Po wpisaniu otrzymamy komunikat np.: **accept male**
male

Teraz wpisujemy nazwę obiektu i przyciskamy klawisz ENTER. W nowej linii ukazuje się kolejne **male.** i system oczekuje na wprowadzenie nowego obiektu lub słowa **end** kończącego wprowadzanie listy obiektów.

accept male
male. (Henry-Snr)
male. (Henry)
male. (Edward)
male. end

Możemy teraz przeczytać nasze dane używając komendy **list all**

&. list all
Henry-Snr father-of Henry
Henry father-of Mary
Henry father-of Elizabeth
Henry father-of Edward
Elizabeth mother-of Henry
Katherine mother — of Mary
Ann mother-of Elizabeth2
Jane mother-of Edward
Henry-Snr male
Henry male
itd.

Możemy wybrać pojedynczą relację i wypisać obiekty spełniające ją. Uczynimy to w następujący sposób: po komendzie **list** wypiszemy nazwę relacji

&. list mother-of
Elizabeth mother-of Henry
Katherine mother-of Mary
Ann mother-of Elizabeth2
Jane mother-of Edward
&

Aby zapisać ten plik danych na taśmie piszemy:

&. save RODZINA

Z powrotem ładujemy do komputera plik rozkazem

&. load RODZINA

Kasowanie lub dopisywanie dowolnych zdań w micro-PROLOG-u jest bardzo proste. Zdanie:

Katherine mother-of Mary

możemy skasować w dwojaki sposób. Pisząc

&. delete (Katherine mother-of Mary)

lub

&. delete mother-of 2

W pierwszym przypadku wskazujemy dokładnie na relację, w drugim czynimy to pośrednio przez wskazanie numeru, pod którym dana relacja się znajduje. Podobnie jeśli przy instrukcji **add** podamy numer, to zdanie, które wpiszemy znajdzie się w odpowiednim miejscu.

&. add. 5 (Katherine mother-of Mary)

Komenda **kill** w połączeniu z nazwą relacji kasuje wszystkie zdania wykorzystujące daną relację.

&. Kill male

Cały program kasujemy przy pomocy **Kill all.** Istnieje także komenda **NEW.** Kasuje ona nie tylko wszystkie dane, lecz i program SIMPLE (przypominam, że SIMPLE to tylko rozszerzenie micro-PROLOG-u ułatwiające pisanie programów). **Po użyciu tej komendy trzeba powtórnie załadować SIMPLE.**

Potrafimy już tworzyć zbiory danych i czytać je. Obecnie nauczymy się zadawać komputerowi pytania tak, by uzyskać na nie odpowiedzi. Nadal będziemy się posługiwali danymi dotyczącymi rodziny. Najprostszą formą pytania jest prośba o potwierdzenie faktu. Pytamy się, czy Henryk jest ojcem Elżbiety 2 ? W PROLOG-u pytanie to zadajemy w sposób następujący:

&. is (Henry father-of Elizabeth2)

na co PROLOG odpowiada:

YES

Odpowiedź na pytanie: is (. . . . . .) polega, po prostu, na sprawdzeniu czy zdanie (. . . . . .) lub inne, rownoważne, figuruje na liście danych. Dużo bardziej skomplikowane jest pytanie typu:

Czy jest znana matka Marii?

W PROLOG-u pytanie takie wygląda następująco:

&. is (x mother-of Mary)

Czyli: czy znany jest taki obiekt **x**, że zdanie: **x mother-of Mary** jest prawdziwe. PROLOG znajduje zdanie: **Katherine mother-of Mary** i wysyła odpowiedź **YES.**

W tym przypadku **x** jest zmienną. Zmienna jest traktowana w PROLOG-u jako obiekt nieznany. Jej odpowiednikiem może być w języku naturalnym na przykład „ktoś" „coś". Zmienne oznaczamy literami **x, y, z, X, Y, Z,** (duże litery oznaczają zbiory). W przypadku większej ilości zmiennych, możemy je indeksować np. **x1, x2, x3...**

Kto jest ojcem Edwarda? W PROLOG-u piszemy:

&. which (x : x father-of Edward)

czyli: znajdź taki obiekt x, że prawdziwe jest zdanie: x father-of Edward.

PROLOG dopuszcza także pytania złożone, np.:

Czy Henryk senior jest ojcem Henryka i Edwarda?

&. is (Henry-Snr father-of Henry
1. and Henry-Snr father-of Edward)
NO

Znak **1** pojawia się po przejściu do nowej linii (naciśnięcie klawisza ENTER) i oznacza, że zdanie nie zostało zamknięte znakiem **)** i może być kontynuowane.

W następnym odcinku trochę arytmetyki.

ADAM KRAUZE
(lat 17)

Zakończyła się krwawa, bezsensowna bitwa przeciwko siłom z Conachta. Cuchulainn Wielki powracał do swej siedziby w Muimethne. Towarzyszył mu wierny powożnik rydwanów Loeg. Bitwa okazała się omyłką, gdyż została stoczona przeciwko malemu oddziałowi zwiadowczemu, a nie ekspedycji wysłanej dla zdobycia wzgórza Beann Ghulban, jak mylnie sądził Cuchulainn. Fatalna była również z innego powodu. Otóż w bitwie poległ Amhair — władca Connachtu, syn suwerena całego kraju. Ojciec przysiągł srogo pomścić pierworodnego.

Drugiego dnia wędrówki Cuchulainn i Loeg zatrzymali się w przydrożnej gospodzie. Zażądali jadła i noclegu. Po chwili podeszła do nich olśniewająco piękna dziewczyna i poprosiła o pomoc w naprawie złamanej osi w jej powozie.

Pierwsza Błyskawica rozerwała skrzydła wiecznej Nocy. Świat ukazał się oczom Człowieka. Głos Ciemności został rozszarpany przez Grom, który przetoczył się w swym majestacie z góry ku dołom, korytami rzek do morza. Symbolem Gromu stał się Dąb rosnący w Świętym Gaju Dun Darach, jego korzenie spajały Ziemię, a na gałęziach spoczywał Firmament. Wzgórze Dębu otoczone zostało największą tajemnicą dostępną jedynie wybranym Druidom.

Z czasem wiedza o Świętym Gaju przeniknęła na zewnątrz. Druidzi wybudowali wokół Wielki Zamek, aby ukryć go przed wzrokiem ciekawskich. Wkrótce dookoła Zamku wyrosło miasto zamieszkane przez wielu ludzi. Tajemnica Świę-

DUN DARACH – ZAKLĘTE MIASTO

NUMER JEDEN NA LISTACH PRZEBOJÓW

Z niezwykłym dla niego pośpiechem, Loeg zaofiarował pomoc. Gdy Cuchulainn wyszedł, by zobaczyć, jakie postępy czyni jego przyjaciel w naprawie osi (i nie tylko), spostrzegł ze zdziwieniem, że Loeg i nieznajoma zniknęli. Nie opadł jeszcze tuman kurzu wzniesiony przez powóz, który znikał daleko na wschodzie, wiatr przyniósł demoniczny śmiech dziewczyny. Cuchulainn, pełen niepokoju, powrócił do karczmy i od przerażonego oberżysty dowiedział się całej prawdy.

Skar była czarodziejką skoligaconą z dworem Connachtu. Uprowadziła Loega z zemsty za śmierć księcia, by uwięzić go w zaklętym mieście Dun Darach.

Cuchulainn przysiągł na wszystkich bogów wydobyć przyjaciela spod władzy czarownicy. Dosiadł rumaka i wyruszył w kierunku tajemniczego Grodu.

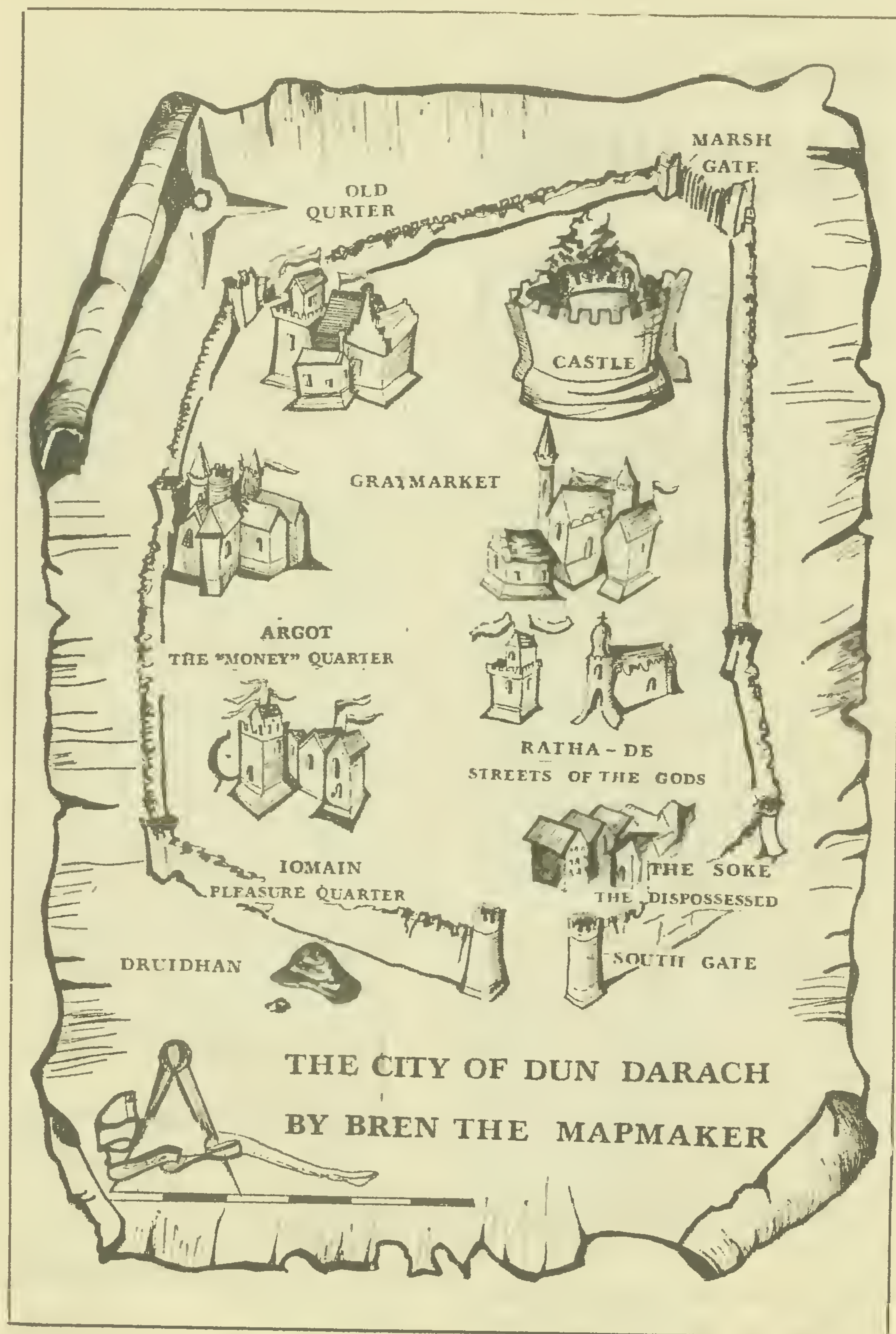
Początki miasta Dun Darach sięgają powstania Świata. Oto relacja anonimowego, starożytnego autora:

tego Gaju znowu została zagrożona. Zdesperowani Druidzi zwołali zgromadzenie największych Magów, którzy rzucili Zaklęcie. Natychmiast zimna biała mgła opadła na Dun Darach. Miasto wraz z okolicą uwolniło się z granic Świata i zaczęło dryfować w czasie.

Rozpoczynając Grę utożsamiamy się z Cuchulainnem stojącym u wrót Zaklętego Miasta. Czeka nas niezmiernie trudne zadanie — odszukać Loega. Nie wszyscy mieszkańcy Grodu powitają nas przyjaźnie. Są wśród nich złodzieje, kurtyzany i płatni czarownicy. Przyjdzie nam nieraz spotykać się z nimi, zdobywać potrzebne przedmioty, informacje... Jedynym pocieszeniem jest fakt, że dla Miasta Czas nie istnieje, jesteśmy więc nieśmiertelni.

Dodajmy na zakończenie, że do pokonania niezliczonych trudności przyda Ci się bardzo znajomość mitologii celtyckiej. Warto więc poczytać.

ADAM KRAUZE


MARSH GATE
OLD QURTER
CASTLE
GRAYMARKET
ARGOT
THE "MONEY" QUARTER
RATHA - DE
STREETS OF THE GODS
IOMAIN
PLEASURE QUARTER
THE SOKE
THE DISPOSSESSED
DRUIDHAN
SOUTH GATE
THE CITY OF DUN DARACH
BY BREN THE MAPMAKER

PASKUDNE MRÓWKI

Użytkownik mikrokomputera, szczególnie tak popularnego jak ZX Spectrum, nie ma już dzisiaj specjalnego kłopotu ze zdobyciem ciekawego oprogramowania. Dlaczego więc tak dużym powodzeniem cieszą się listingi zamieszczane w różnych pismach? Przecież nawet najprostszy program fabryczny jest bardziej atrakcyjny niż ten, który jesteśmy w stanie sami sobie „wpalcować".

Myślę, że tajemnica tkwi w fakcie, iż listing pozwala nam na pewną samodzielność. Własnoręczne wpisanie programu umożliwia analizę jego działania, wprowadzanie pewnych zmian (jak chociażby komentarze w języku polskim), uzupełnianie lub upraszczanie jego funkcji. Oczywiście bardziej zaawansowani amatorzy informatyki mogą to samo robić z programami fabrycznymi, lecz dla początkujących jest to próg nie do przekroczenia.

Dlatego właśnie zdecydowaliśmy się drukować w BAJTKU listingi, rzecz jasna postaramy się wybierać najciekawsze. Zacznijmy od zabawy — gry pt. Piknik. Oczywiście, komputer w tej grze przemawia do ciebie po polsku i w dodatku ortograficznie (to znaczy używa liter takich jak „ó" czy „ą").

Oto jej treść: wybrałeś się na uroczą — jak ci się zdawało — wycieczkę na łono natury. Niestety, stałeś się ofiarą napadu ze strony okrutnych mrówek, które usiłują porwać ci śniadanie. Jeśli nie chcesz być głodny, musisz zdeptać złodziejki. Twoje zapasy składają się z trzech owoców, zaś każde cztery mrówki, które przepuścisz uniosą jeden z nich. Staraj się więc... Jest jeszcze jedna trudność, ale o niej dowiesz się sam, gdy uda ci się uruchomić program.

Gra ta, opracowana przez pana Grahama Ramsdena (z drobnymi przeróbkami piszącego te słowa) nie wzbudzi zapewne entuzjazmu wśród miłośników przyrody. Ma jednak kilka zalet, niezależnie od tego, kto kogo w niej depcze.

Przy pomocy stosunkowo krótkiego programu, udało się autorowi uzyskać ciekawą grafikę i animację. Gra uzupełniona jest również zupełnie przyzwoitymi efektami dźwiękowymi. Tym, którzy mieliby ochotę spróbować sił w konstruowaniu prostych gier polecam, by przeanalizowali sposób uzyskiwania obrazów poszczególnych obiektów. Są one definiowane jako znaki graficzne i drukowane instrukcją PRINT. Jak to się natomiast dzieje, że komputer wyświetlając na przykład mrówkę w kolejnym położeniu, wygasza jednocześnie jej poprzedni wizerunek? Spróbuj sam odpowiedzieć na to pytanie.

Jeśli znajdujesz się na początkowym etapie komputerowego wtajemniczenia, możesz mi wierzyć, że warto poświęcić trochę czasu na zrozumienie tego programu. Zwróć — między innymi — uwagę jak zostały uzyskane polskie litery „ó" i „ę". Zrozumiesz, że wykorzystując zastosowane tu sposoby, możesz własnoręcznie napisać podobny program.

Opis funkcji poszczególnych linii

7 — 8	deklaracja zmiennych
9	wstępna plansza gry
10 — 40	wczytanie określenia znaków graficznych, opisanych pod etykietą DATA
50	skok do podprogramu drukującego opis gry
60	skok do podprogramu drukującego nagłówek gry
70	rysowanie trawnika
80 — 82	skoki do odpowiednich podprogramów w zależności od liczby owoców, które pozostały grającemu
90 — 100	odczytywanie znaków wprowadzanych z klawiatury
120 — 130	animacja mrówek
140	sprawdzenie, czy mrówka doszła do owocu i liczenie tych mrówek
145	sprawdzenie czy już cztery mrówki dotarły do owoców, jeśli tak to zmniejszenie liczby owoców o jeden i skok do 81
150	aktualna punktacja
155	sprawdzenie czy są jeszcze owoce, jeśli nie to skok do 4030
160	powrót do 90
200 — 220	animacja nogi, sprawdzenie czy nie nadepnąłeś motyla, czy nie nadepnąłeś mrówki, wyświetlanie i znikanie motyla
1000 — 1030	nagłówek gry
2000 — 2030	napis kończący grę
3000 — 3096	zasady gry
4000 — 4010 4015 — 4020 4030 — 4040	animacja mrówek niosących owoce

Piknik - gra na ZX Spectrum
16/48 kB
A - klawisz A w trybie graficznym
G1 - klawisz 1 w trybie graficznym
SG1 - klawisz 1 + SHIFT w trybie graficznym

```
7 LET a=10 : LET poz=0: LET pkt=0:
LET owoce=3: LET mr=0
8 LET a$="AB":LET b$="CD":
LET c$="EF": LET d$="GH":
LET e$="IJ": LET f$="KL":
LET g$="RS"
9 CLS : LET m$="  P I K N I K '
FOR q=19 TO 1
STEP -2: PRINT AT q,6:
INK 0; PAPER 7; BRIGHT 1;
FLASH 1;m$: BEEP .01,q+10:
BEEP .01,q+13: BEEP .01.q+16:
NEXT q: LET q$="Uwaga na motylka!!!"
: FOR c=1 TO LEN q$: LET r$="":
LET r$=r$+q$ (c) : PRINT
AT 10,c+5; INK 6; PAPER 2;
BRIGHT 1; FLASH 1;r$:
BEEP .01,c: BEEP .01,c+3:
BEEP .01,c+6: NEXT c:
BRIGHT 0
10 FOR f=0 TO 20: FOR n=0 TO 7
: READ x: POKE USR CHR$ (144+f)+n,x:
NEXT n: NEXT f
20 DATA 0,0,0.0,1,3,3,3,0,96,144,
128,128,192,192,192,7,7,15,31,29,
24,13,7,224,224,240,248,248,248,
240,224,30,15,7,3,3,1,1,1,0,128,
224,240,248,252,252,252,1,1,1,3,3,
7,15,30,252,252,252,248,240,224,128,
0
30 DATA 6,1,6,1,15,17,44,45,32,64,
176,64,240,136,52,180,49,16,27,11,
8,7,3,1,140,8,104,112,16,96,64,128,
0,0,227,236,82,137,0,0,0,0,192,199,
55,74,145,0,153,153,153,153,153,153,
153,153,0,1,3,43,85,234,255,127,126,
36,36,194,199,255,255,254,36,50,105,
125,62,53,25,17,36,76,150,190,124,
252,152,136,
40 DATA 0,0,56,4,60,68,60,6,8,16,56,
68,68,68,56,0
50 GO SUB 3000
60 GO SUB 1000
70 PRINT AT 21,0; INK 4;">32SG8< "
80 IF owoce=3 THEN PRINT AT 19,29:
INK 4;a$;AT 20,29;b$
81 IF owoce=2 THEN GO SUB 4000
82 IF owoce=1 THEN GO SUB 4015
90 LET a=a+(INKEY$="0" AND a<2
6) - (INKEY$="9" AND a>0) :
PRINT AT 2,a; INK 1;" PQ '
100 IF INKEY$="1" THEN
GO SUB 200
120 PRINT AT 20,poz: INK 0;" M"
: BEEP .001,50: BEEP .001,4
0: BEEP .001,30
130 LET poz=poz+1
140 IF poz=28 THEN LET mr=mr+1:
LET poz=0: PRINT AT 20,31:
INK 0;"N";AT 20,28;" "
145 IF mr=4 THEN LET owoce=owoce-1:
BEEP .25,30: BEEP .75,0:
LET  mr=0: PRINT AT 20,poz;" ":
LET poz=0: PRINT AT 20,31;" ":
FOR p=2 TO 16:
PRINT AT p,0; INK 0;"

: BEEP .01,p+10: NEXT p:
NEXT n: GO TO 81
150 INK 1: PRINT AT 0,10: pkt:
PRINT AT 0,29; owoce
155 IF owoce=0 THEN GO TO 4030
160 GO TO 90
200 FOR f=2 TO 14: PRINT AT f+1
,a+1; INK 1;"PQ";AT f,a+1;"
O":: BEEP .001,f+20:
BEEP .001,f+22: BEEP .001,
f+24: NEXT f: IF ATTR (f+1,a+1)=43 OR
ATTR (f+1,a+2)=43  THEN LET mr=4:
IF mr=4 THEN GO TO 145
201 FOR z=15 TO 18: PRINT AT z+1,
a+1; INK 1;"PQ";AT z,a+1; " O ":
BEEP .001,z+20:
BEEP .001,z+22: BEEP .001,z+24:
NEXT z
205 IF z+1=20 AND a+1=poz
THEN LET pkt=pkt+2: PRINT
AT 20,poz; INK 0:"v";AT 20,poz;
OVER 1;"x": FOR i=0
TO 4: BEEP .01,55: BEEP .01,53:
BEEP .01,51: NEXT i: LET poz=0
206 IF z+1=20 AND a+2=poz
THEN LET pkt=pkt+4: PRINT
AT 20,poz; INK 0;"x";AT 20,
poz; OVER 1;"v": FOR i=0
TO 4: BEEP .01,55: BEEP .01,53:
BEEP .01,51: NEXT i: LET poz=0
210 FOR f=20 TO 3 STEP -1:
PRINT AT f,a+1;" ";AT f-1
,a+1; INK 1;"PQ": BEEP .001,f+20:
BEEP .001,f+22: BEEP .001,f+24:
NEXT f
211 PRINT AT 16,INT (RND*10)+2;"
215 PRINT AT 16,a+1; INK 3;g$
220 RETURN
1000 BORDER 5: PAPER 5: INK 5:
CLS
```

```
1010 PRINT INK 1;"Punktacja =";pkt">SG8<"
PIKNIK>SG8<owoce=";owoce:";owoce;" M"
1030 FLASH 0: RETURN
2000 PRINT AT 10,5 INK 3;
FLASH 1: PAPER 7; "Koniec twojego pikniku"
2010 FOR n=50 TO 0 STEP -1:
BEEP .1,n: BEEP .1,n+2:
BEEP .1,n+4: NEXT n
2020 FOR v=50 TO 30 STEP -1:
BEEP .001,v+6: BEEP .001,v+3:
BEEP .001,v: NEXT v
2030 RUN
3000 PAPER 4: INK 0: CLS
3010 PRINT " IJ N    P I K N I K
M IJ
3040 PRINT "RSRSRSRSRSRSRSRSRSRSR
SRSRSRSRSRS"
3050 PRINT INK 7:";"
Cichy, spokojny piknik...nagle
nadchodzT okrutne mrUwki i kradnT twöje
owoce."
3060 PRINT ": INK 7:"    ST bardzo silne
, cztery  mrUwki same zaniöeT owoc do
swojej kryjUwki."
3070 PRINT ";INK 7:"    Rozdeptaj je, bo
nic ći nie zostawiT."
3075 PRINT "INK 7;"      9 .:.. w lewo"
;"":"     0 ... w prawo";""      1
... deptanie"
3080 FOR g=0 TO 5: FOR l=0 TO 7:
BEEP .1,l+40: BORDER l
3081 IF INKEY$<> "" THEN RETURN
3082 NEXT l
3085 FOR k=1 TO 0 STEP -1: BEEP .1,k+40:
BORDER k
3090 IF INKEY$<> "" THEN RETURN
3095 NEXT k: NEXT g
3096 CLS : RETURN
4000 FOR x=28 TO 1 STEP -1: PRINT AT
18,x: INK 4;"AB ";AT 19,x;"CD ":AT
20,x-1;INK 0:"N N ": BEEP .01,x:
BEEP .01,x-2: BEEP .01,x-4: NEXT x:
PRINT AT 18,0:"    ";AT 19,0:"    "
4001 PRINT AT 19,29: INK 6;c$: AT 20,29
;d$
4002 LET owoce=2
4010 RETURN
4015 FOR x=28 TO 1 STEP -1: PRINT AT 18,x
; INK 6; "EF ";AT 19,x;"GH ":AT 20,x-1:
INK 0;"N N ": BEEP. 01,x-4:NEXT x: PRINT
AT 18,0:"    ";AT 19,0:"    "
4016 PRINT AT 19,29; INK 2;e$: AT 20,29
;f$
4020 RETURN
4030 FOR x=28 TO 1 STEP -1: PRINT
AT 18,x: INK 2;"IJ ";AT 19,x;"KL ";AT
20,x-1;INK 0;"N N ": BEEP .01,x-4
:BEEP .01,x-2: BEEP .01,x-4: NEXT x:
PRINT AT 18,0:"    ";AT 19,0:"    "
4040 GO TO 2000
```

Przyjemnej zabawy!

ROMAN POZNAŃSKI

BASIC w TV

KORESPONDENCJA Z WĘGIER

Komputer traktuje się tu w kategoriach węgierskiej szansy i nadziei. Nadzwyczaj ulgowo obchodzi się z nim obowiązująca niemal każdą dziedzinę życia zasada „zaciskanego pasa". Nawet biedny na ogół resort oświaty ma środki na zaprzyjaźnienie go z najmłodszymi Węgrami.

Myślące maszyny są już w każdej węgierskiej szkole średniej, a od września — w 300 podstawowych. **Ćwierć miliona osób przed ekranami telewizyjnego kursu techniki obliczeniowej, dziesiątki wakacyjnych turnusów z komputerem w roli wychowawcy, obniżone cło na przewóz z zagranicy osobistych maszyn liczących do prywatnego użytku — to zaledwie część działań, w efekcie których Węgrzy nie chcą stracić kontaktu z techniką światową.**

Do tego co się dzieje na komputerowym monitorze zdecydowana większość społeczeństwa ma wciąż stosunek nieufny. Ktoś, kto sam nigdy nie naciskał klawiszy tajemniczej aparatury — pojawiające się na ekranie rzędy cyfr, wykresów i napisów odbiera w kategoriach fantastycznej bajki naukowej. Doświadczenia z uczniami starszych klas dowiodły, że nie taki diabeł straszny, że warto i trzeba — jeżeli poważnie traktuje się całe przedsięwzięcie — komputer ustawić obok elementarza. Ale jak przekonać do sprawy starszych?

Towarzystwo Naukowe Techniki Obliczeniowej im. Janosa Neumanna (zmarły w 1957 roku w Waszyngtonie światowej sławy matematyk węgierski) zapukało do drzwi rodzimej telewizji, a ta szczęśliwie kupiła pomysł zorganizowania „powszechnego nauczania techniki obliczeniowej". Początkowo adresatami miały być osoby powyżej trzydziestu lat, chodziło bowiem o podjęcie próby przełamania wątpliwości u dorosłych, u tych, którzy nie wierzą, by człowiek w starszym wieku mógł przyswoić sobie zupełnie nową technikę. Po przyłączeniu się trzeciego partnera — Przedsiębiorstwa Eksploatacji Maszyn Liczących (SZAMALK) — w styczniu br. poszła w eter pierwsza lekcja półrocznego kursu TV—BASIC.

Zgodnie z węgierską tradycją rzecz całą znakomicie zorganizowano. Inauguracyjny wykład poprzedziło bardzo staranne przygotowanie gie-

by, by — mówiąc gornolotnie — rzucone weń ziarno wiedzy jak najlepiej zaowocowało. Wydano więc książkę dr. Andrasa Kocsisa, bez lektury której telewizyjne wykłady traciłyby sens. Ta teoretyczna podstawa nauczania ukazała się w nakładzie 100 000 egzemplarzy, co w tych warunkach stanowi ogromną ilość, a mimo to trudno ją było dostać. By uczestnicy kursu mieli gdzie i na czym odrabiać lekcje i ćwiczyć — Magazyn Mikrokomputerowy otworzył ponad 300 klubów, zaś Towarzystwo im. Neumanna — w każdym województwie i w kilku wojskowych szkołach.

Telewizyjny Uniwersytet Komputerowy znalazł ponadto sojuszników w Ministerstwie Kultury, w Towarzystwie Krzewienia Wiedzy Przyrodniczej, Instytucie Oświaty Ludowej i KISZ. Ci zaś, wydając stosowne zarządzenia, polecili swoim placówkom stale udostępniać posiadane maszyny liczące kursantom. KISZ-owski tygodnik twórczej młodzieży „Oetlet" na kilka miesięcy zamienił się w prasowy organ TV—BASIC, systematycznie publikując kolejne lekcje. Wykładowcy cały czas mieli na uwadze, by zadania nadawały się do rozwiązywania na czterech rozpowszechnionych na Węgrzech typach urządzeń: komputerze Przedsiębiorstwa Techniki Łączności, upowszechnionym w szkołach, „Commodore", „Sinclair Spectrum" i „Primo".

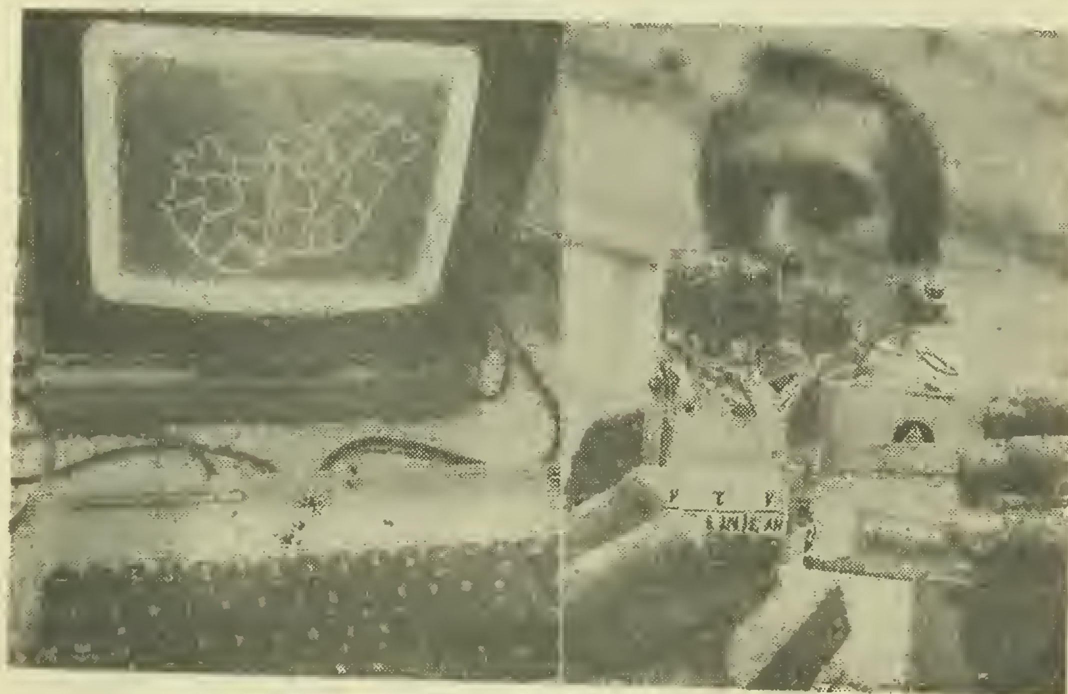
Blisko półroczną edukację zakończył egzamin. Świadectwo wydawano po udzieleniu 80 procent prawidłowych odpowiedzi na 74 pytania. Nie pobłażano, skoro wiele przedsiębiorstw postanowiło je traktować jako dowód umiejętności z dziedziny programowania. Ponadto miało być ono warunkiem przyjęcia na kurs „SZAMALK" wyższego stopnia. Dodatkowe nagrody w postaci komputerów „Primo" czekały na 2 jednostki wojskowe, z których najwięcej żołnierzy pomyślnie przejdzie egzaminacyjną próbę.

Do kursu TV—BASIC szybko przylgnęło określenie „bezprecedensowego" w węgierskim systemie oświaty". Według Ośrodka Badawczego Komunikacji Masowej, kilkumiesięczne wykłady śledziło tyle osób co drugie wydanie Dziennika Telewizyjnego, czyli 240—250 tysięcy Węgrów. Do egzaminu stanęło 2360 mężczyzn i 439 kobiet, choć wcześniej chęć zadeklarowało ponad 6350 osób. Najmłodszy miał 8 lat, najstarszy — 78.

Ośmiolatek zgromadził 58 punktów i zdał, dwóch 78-letnich dziadków, niestety, odpadło. Najstarsi spośród tych, którzy zdali, mieli po 66 lat. Najliczniejszą grupę wśród egzaminowanych stanowiła młodzież w wieku 14—19 lat, natomiast najwyższe noty zebrali: 27-letnia mieszkanka Pecsu i 17-latek z Dombovaru. W sumie wymagane minimum (57 punktów) uzyskało 44 procent zdających, co przy tego rodzaju systemie nauki uchodzi za wynik dobry.

— Nie tylko my, ale i nasi „słuchacze" — mowi Gyoezoe Kovacs, sekretarz generalny Towarzystwa Naukowego Techniki Obliczeniowej im Neumanna — życzą sobie kontynuowania podobnego nauczania, co potwierdzają w licznych listach i rozmowach. Pracujemy więc nad materiałami do dwóch następnych kursów, nad odpowiednią książką i zestawem telewizyjnych audycji. Pierwszy to „Wprowadzenie do zastosowania techniki obliczeniowej" drugi — „Technologia programowania". Start chyba jesienią, egzaminy — wiosną przyszłego roku.

GRZEGORZ ŁUBCZYK



Fot. Michał Kułakowski

Roger Zelazny

SEKSKOMPUTER

Nie wszyscy wiedzą, że niektóre maszyny uprawiają miłość i dzieje się tak wbrew metafizycznym rozważaniom Świętego Jakuba Mechanofila. Święty Jakub Mechanofil uważa bowiem człowieka za organ seksualny maszyny, która go stworzyła i sądzi, że istnienie człowieka jest niezbędne dla wypełnienia się misji życiowej tegoż mechanizmu. Twierdzi on, że maszyna wytwarza kolejne generacje maszynotypów, a wszystkie stadia owej mechanicznej ewolucji przepływają przez człowieka aż do momentu, w którym osiągnięta zostaje absolutna doskonałość produktu i człowiek, zrobiwszy swoje, może poddać się Wielkiej Kastracji.

Święty Jakub jest, rzecz jasna, heretykiem. Jak zostało udowodnione w wypadkach zbyt licznych na to, by je tu przytaczać, maszyna jako całość musi dysponować płcią. Ponieważ człowiek i maszyna wymieniają się często składnikami lub nawet całymi systemami, złączone razem stają się tworem, który może zaistnieć w dowolnym miejscu mechaniczno-człowieczego



spektrum i przebyć całość cyklu. Dlatego człowiek, ów organ zarozumiały, osiągnął swą apoteozę, czy też doskonałe zespolenie, z Uszlkotłokiem poprzez ofiarę i pokutę. Wchodzi tu również w rachubę pomysłowość, lecz pomysłowość sama w sobie jest, oczywiście, formą mechanicznej inspiracji. Nie można więc już dłużej traktować idej Wielkiej Kastracji, rozważać możliwość odłączenia maszyny od tego, co ją stwarza. Człowiek na zawsze pozostanie częścią Wielkiego Obrazu.

Wszyscy wiedzą, że maszyny uprawiają miłość. Oczywiście nie w sensie dosłownym, inaczej niż to robią kobiety i mężczyźni, którzy, pomijając tu rodzaj pobudek ekonomicznych, jakimi się kierują, odnajmują swe ciała na rok czy dwa jednemu z dobrych towarzystw handlowych, pozwalając zespolić się z maszyną, karmić dozylnie, gimnastykować izometrycznie, pogrążyć się w podświadomość (lub zachować pełnię władz umysłowych jeżeli tego sobie życzą), narażają mózg na implantacje stymulujące odpowiednimi ruchami przez okres nie dłuższy niż piętnaście minut za jedną monetę, dają się umieszczać na kanapach ekstrawaganckich klubów (coraz częściej w zamożniejszych domach, ale też i w tanich budkach na rogach ulic) ku uciesze i dla rozrywki swych pobratymców. Nie. Maszyny kochają się via człowiek, lecz ponieważ opisywano już przypadki przeniesienia funkcji, robią to zazwyczaj odruchowo.

Zajmijmy się teraz jedynym w swoim rodzaju fenomenem, który dopiero co pojawił się na rynku: Sekskomputer — komputer—wyrocznia, który udziela odpowiedzi na pytania z każdego zakresu i odpowiada tak długo jak długo pytający zdoła go utrzymać w stanie odpowiedniego podniecenia. Ilu to z was zdążyło już odwiedzić ten zaprogramowany buduar, aby rozważyć, rozstrzygnąć jakiś problem olbrzymiej wagi? Ilu to z was zauważyło, jak szybko mija tam czas...? Właśnie. Wygląda niczym anty-centaur (człowiek od pasa w dół), jest wcieleniem wszystkiego, co najlepsze u maszyny i u istoty ludzkiej i jest symbolem doskonałej fuzji tych dwóch form. Jak zawsze w takich wypadkach, powstała już rzewna historyjka miłosna o mężczyźnie, który wchodzi do Kabiny Pytań, żeby wywiedzieć się o losy swej ukochanej. A maszyna mu na to, że... — nie, o tym później. Dzieje się tak przecież zawsze i wszędzie i nierzadko nie znajdujemy nic równie wzruszającego (...)

MIŁOŚĆ PRZYSTOI HORUSOWI

Przemierza ulice światła i nocy i nikt nie zna tu jego imienia. Jeżeli jednak zapytać o to którąkolwiek z przechodzących istot, odpowiedziałaby, że nie jest jej obce. Jest bogiem i siła jego przeogromna. A mimo to uległ Księciu, swemu bratu, który go zgubił, ratując własne życie i bieg rzeczy, którego jest uosobieniem.

Horus skręca obecnie w dobrze oświetloną alejkę, gdzie kłębią się przeróżne stwory. Otacza go moc i jasność.

Konkretna ulica, konkretny świat — nie przybył tu bez powodu: Horus wciąż się waha i chce wysłuchać zdania innych. Uwielbia wróżby.

Szuka rady.

Czerń nieba i potoki światła zabawy — Horus mija jaskinie rozrywki i rozochoconych klientów.

Jakiś człowiek zagradza mu drogę. Horus próbuje go ominąć i zbacza na ulicę. Człowiek idzie za nim i chwyta go za rękaw.

Bóg ledwie chucha na niego, a tchnienie ma siłę huraganu. Porywa tamtego, zmiata go i Horus rusza dalej.

Po chwili znajduje się w dzielnicy wróżbitów. Astrolodzy, zwolennicy kart Tarota, odczytujący przepowiednie z liczb i z Księgi Przemian I Cing kiwają na boga w czerwonej przepasce, lecz on nie zatrzymuje się.

Aż dochodzi tam, gdzie nie ma ludzi. Stoją tu maszyny, które wróżą.

Na chybił trafił wybiera kabinę i otwiera ją.

— Czym mogę służyć? — odzywa się kabina.

— Zapytania — rzuca Horus.

— Chwileczkę.

Słychać metaliczny szczęk i otwierają się ukryte drzwi.

— Proszę wejść do środka.

Horus wchodzi do maleńkiego pokoiku. Nie ma tu nic z wyjątkiem czegoś w rodzaju łoża. Na łożu leży kobiecy tułów podłączony do błyszczącej konsoli. W ścianie tkwi wbudowany głośnik.

— Proszę zająć miejsce w zespole informacyjnym — słyszy.

Zrzuciwszy przepaskę, Horus postępuje według instrukcji.

— Przepisy mówią, że na pytania będzie się odpowiadać dopóty, dopóki sprawiać mi będziesz rozkosz. Cóż chcesz wiedzieć?

— Mam problem. Popadłem w konflikt z bratem. Chciałem go pokonać, lecz nie udało się. Nie mogę się zdecydować czy powinienem go znów odszukać i spróbować jeszcze raz, czy...

— Brak wystarczających danych — pada w odpowiedzi. — Jaki to konflikt? Jakiego rodzaju brat? Do jakiego gatunku ludzi należysz?

Smutno kwitnie bez, a rożane pasaże to cierniste żywopłoty. Ogród wspomnień mieni się oszalałymi bukietami...

— Chyba źle trafiłem...

— Może tak, może nie. Z pewnością jednak nie znasz przepisów.

— Przepisów? — Horus spogląda na nieprzeniknioną ścianę głośnika.

Głośnik cedzi suchym, monotonnym głosem:

— Nie jestem jasnowidzem, ani wróżbitą. Jestem elektro-mechaniczno-biologicznym wyznawcą boga Logiki. Moją ceną przyjemność i za tę cenę odwołam się do boga dla każdego mężczyzny. Aby to jednak uczynić, muszę dysponować bardziej konkretnymi pytaniami. Jak dotąd nie dostarczyłeś mi wystarczających danych. Kochaj mnie więc i powiedz coś więcej.

— Nie wiem od czego zacząć... — mówi Horus. — Kiedyś mój brat władał wszystkim...

— Stop! Zdanie nielogiczne, nieprecyzyjne...

— ...i absolutnie poprawne. Moim bratem jest Tot, zwany niekiedy Księciem, Który Ma Tysiąc Lat. Wszystkie Światy Wewnętrzne należały do jego królestwa.

— Zapis mojej pamięci wskazuje na istnienie mitu o Władcy Życia i Śmierci. Zgodnie z przekazem on nie miał braci.

— Stop! Poprawka. Takie sprawy nie wychodzą z reguły poza rodzinę. Izyda miała trzech synów: jednego z prawego łoża, od Ozyrysa, dwóch spłodził Set Destruktor. Setowi urodziła Tyfona i Tota. Ozyrysowi — Horusa-Msciciela, czyli mnie.

— Tyś Horus?

— Rzekłaś.

— Chcesz zabić Tota?

— Tak. Takie otrzymałem polecenie.

— Nie możesz tego zrobić.

— Ach tak...

— Nie odchodź, proszę. Może chcesz mi zadać jakieś inne pytania?

— Nic mi nie przychodzi do głowy.

Horus nie może jednak odejść, gdyż ogarnia go ogień.

— Kimże jesteś? — pyta w końcu.

— Już ci mówiłam.

— Ale jak się to stało, że jesteś tym, czym jesteś — pół kobietą, pół maszyną?

— To jedyne pytanie, na które nie mogę odpowiedzieć bez niezbędnych dyspozycji. Aczkolwiek spróbuję cię pocieszyć, gdyż widzę, żeś w rozterce.

— Dziękuję. Miła jesteś.

— Cała przyjemność po mojej stronie.

— Byłaś niegdyś człowiekiem, czyż nie tak?

— To prawda.

— Dlaczego przestałaś nim być?

— Pytałeś już. Nie mogę odpowiedzieć.

— Czy mogę ci pomóc i wpłynąć jakoś na to, czego pożądasz?

— Tak.

— W jaki sposób?

— Nie mogę odpowiedzieć.

— Czy jesteś stuprocentowo pewna, że Horus nie jest w stanie pokonać Tota?

— Moja znajomość legend mówi, że to wysoce nieprawdopodobne.

— Gdybyś była zwykłym śmiertelnikiem skłonny byłbym okazać ci dobroć.

— Co to znaczy?

— Kochałbym cię za twą przerażającą szczerość.

— Boże! Bój Boże! Zbawiłeś mnie!

— Co chcesz przez to powiedzieć?

— Byłam skazana egzystować tak dopóty, dopóki ktoś, kto jest czymś więcej niźli człowiekiem tylko, spojrzy na mnie z miłością.

— Właśnie tak bym na ciebie patrzył. Czy uważasz, że to niemożliwe?

— Tak, bom zbyt zużyta.

— A więc nie znasz jeszcze Horusa.

— Ależ prawdopodobieństwo takiego zdarzenia jest równe zeru!

— Nie mam nikogo, więc kocham ciebie.

— Bóg Horus mnie kocha?

— Tak.

— Tyżeś mym Księciem i wreszcie się zjawiłeś!

— Nie rozu...

— Uzbrój się ino w cierpliwość — wiele jeszcze się zdarzy.

— Poczekam — mówi Horus i wstaje.

tłum. Anna Kraśko

Fragmenty książki Rogera Zelaznego „Stwory Światła i Ciemności" przygotowywanej do druku przez redakcję fantastyki KAW.

GIEŁDA

INFORMATYKA NA PERSKIM

Skutkiem wzrostu popularności mikrokomputerów w naszym kraju stała się konieczność powstawania infrastruktury handlowo-usługowej, związanej z zakupem i użytkowaniem tego rodzaju sprzętu. Nie da się ukryć, że najszybciej zareagował na tę potrzebę warszawski Jarmark Perski, przeniesiony przed kilku miesiącami z Marymontu na Koło.

Oferta marketingowa Jarmarku jest bardzo szeroka; od samych mikrokomputerów i współpracujących z nimi urządzeń peryferyjnych poprzez różnorodne oprogramowanie, do usług serwisowych, takich jak naprawy i przeróbki sprzętu fabrycznego. Nie ma także kłopotu z wszelkiego rodzaju literaturą, najczęściej jednak w języku angielskim. Ostatnio pojawia się coraz więcej tłumaczeń. Niestety, nie wszystkie mógłbym polecić z czystym sumieniem, gdyż ich jakość jest bardzo różna. Część z nich jest dziełem ludzi znających prawdopodobnie język, nie bardzo jednak orientujących się w sprawach informatyki.

Ceny tego rodzaju wydawnictw wahają się w granicach od 1,5 do 4,5 tys. zł., z wyraźną jednak tendencją zbliżania się do górnego pu-

O SZTUCE KUPOWANIA KOMPUTERÓW

Szwajcarski tygodnik „Die Weltwoche" (nr 19/1985) udziela garści porad jak kupić odpowiedni komputer do użytku osobistego.

Pół kilo spaghetti albo lampę na stół kupuje się bez większego namysłu i nie musząc mieć gwarancji, że dostawca będzie istniał nadal. W przypadku komputera, jednym z kryteriów wyboru jest solidność i gwarancja dalszego istnienia zarówno producenta, jak i dostawcy.

Zatem pierwsza rada: Przed decyzją zakupu należy zasięgnąć informacji o producencie.

Jeszcze ważniejsze jest zadanie sobie samemu pytania: co mi jest właściwie potrzebne? Co chcę osiągnąć za pomocą urządzenia, które zamierzam nabyć?

...du. Nie jest to mało, nawet dla dobrze uposażonych hobbystów, cóż jednak zrobić, jeśli jest to jedyne dostępne źródło?

Wśród oferowanych do sprzedaży komputerów przeważają dwa modele: ZX Spectrum i Commodore 64. Za nowe Spectrum wyposażone w pamięć o pojemności 48 kB, większość sprzedających żąda 90 tys. zł, zaś za jego 16 kB wersję — od 65 do 75 tys. zł. Coraz częściej pojawia się nowsze dziecko pana Sinclaira, czyli ZX Spectrum+. Jego cena jest jednak odpowiednio wyższa, wynosi około 130 tys. zł.

Commodore 64 pozostaje nadal „na fali", zarówno pod względem zainteresowania, jak i ceny, dochodzi ona do 185 tys. za nowy komputer, wyposażony w oryginalny magnetofon. Egzemplarz używany i bez dodatkowego wyposażenia przy odrobinie szczęścia można kupić za 150 tys. zł a nawet taniej. Kłopot jest natomiast z nabyciem fabrycznego magnetofonu, można za to dostać układy dopasowujące „domowej produkcji" i takież magnetofony za 20 tys. zł.

Rzadziej pojawiają się inne wersje Commodora. Model VC 20, również bez magnetofonu, kosztował 50 tys. zł. Jego zaletą jest prawie profesjonalna klawiatura, wadą natomiast — niewielka pamięć.

Niezbyt często pojawia się jeden z lepszych i równocześnie tańszych komputerów domowych — Atari 800 XL. Jego cena (jeszcze przed sprowadzeniem tego modelu do Pewexu) wynosiła 120 tys. zł (bez magnetofonu).

Na „Perskim" można również kupić urządzenia peryferyjne, takie jak drukarki czy stacje dysków. Za kolorową drukarkę do Commodora żądano 150 tys. zł, natomiast za stację dysków do tego samego komputera — 200 tys. zł.

Oczywiście jest i software. Najtańszy program można dostać za 100 zł, cena najdroższych nie przekracza 500 zł. Wybór ogromny. Na życzenie klienta programy mogą być nagrywane na jego własnym magnetofonie. Nasz klient, nasz pan...

*

Jak długo jeszcze „Jarmark" pozostanie Mekką amatorów małej informatyki? Czy wytrzyma konkurencję Pewexu? Przypuszczam, że jego pozycja pozostanie nie zagrożona co najmniej przez kilka najbliższych miesięcy. Z tego też powodu, postanowiliśmy informować naszych czytelników w kolejnych numerach BAJTKA o aktualnych cenach i nowościach na tym rynku.

ROMAN POZNAŃSKI

P.S. Podawane ceny dotyczą pierwszej połowy września.

Oto rada numer dwa: Trzeba sobie jasno uświadomic własne potrzeby, zanim w ogole wejdzie się do sklepu z komputerami.

Następne czarodziejskie słowo to „compatibility", tzn. możliwość przystosowywania komputera i związana z tym „interface", tzn. możliwość przyłączania. Zarówno „przystosowalność", jak i „przyłączalność" rozciągają się na wiele elementów komputera: sam procesor, urządzenia peryferyjne oraz nabyte czy opracowane programy. Jest to ważne, jeśli się chce później wymienić lub rozbudować swoje urządzenie.

Rada numer trzy: Należy się zorientować w tzw. standardzie przemysłowym (określa go chyba nadal firma IBM).

Pozostaje problem pamięci zewnętrznych. W komputerach osobistych dominuje dyskietka (floppy disc); na drugim miejscu znajdują się twarde dyski (hard discs). W zastosowaniach domowych popularne są tasmy magnetyczne w kasetach.

Rada numer cztery: Apetyt rośnie w miarę jedzenia, dlatego nie należy zbyt oszczędnie wyliczać pojemności pamięci.

Przed wielu laty autor opisywanego artykułu propagował UBM, „Universal Business Machine", urządzenie wielofunkcyjne, służące poza wszystkim także do parzenia kawy.



Tu dochodzimy do rad numer pięć i sześć: Nie istnieje uniwersalny komputer zaspokajający równocześnie wszystkie wymagania. Trzeba więc zdecydować się na jedno lub kilka zastosowań. Nie sięgać od razu do gwiazd, lecz świadomie dążyć drogą z dołu do góry.

To jest najtrudniejsze. Wszyscy marzymy o rozwiązaniu maksymalistycznym i zapominamy o zasadach ekonomicznych, a przecież chodzi tu o naszą własną kieszeń.

Ostatnia rada: Kto teraz kupuje PC, jest „w branży". Kto odkłada tę inwestycję, znajdzie jutro jeszcze wydajniejsze produkty, ale będzie mu brakowało dzisiejszego praktycznego doświadczenia.

HISTORIA ZŁOTEGO JABŁUSZKA

DOKOŃCZENIE ZE STR. 4

się zdobyć prawie za bezcen. Byłem więc tym pytaniem lekko zaskoczony, lecz Steve po prostu zadzwonił do producenta i namówił go na wypożyczenie nam próbek tego towaru. Słysząc to podskoczyłem do sufitu. To było cudowne: osiem tych nowych kości pozwalało zastąpić 32 stosowane dotychczas. Było to co prawda nieco skomplikowane, gdyż wymagało multipleksowania adresów, tym samym jednej lub dwóch dodatkowych kości, lecz i tak moje szczęście nie miało granic. Nowe kości były zgodne ze standardem TTL, tak więc mogłem oszczędzić na płytce masę miejsca, dzięki możliwości użycia znacznie mniejszych części, a jednym z moich celów było zbudowanie tak małego komputera, jak to tylko okaże się możliwe. W końcu dopiąłem swego: miałem komputer, który mieścił się na płytce o rozmiarach w przybliżeniu 20 na 25 cm. Mogłem więc wziąć go w rękę i pójść do klubu, by pochwalić się mym dziełem, Zbudowany był on jedynie z 30 do 40 kości i radził sobie z BASIC-em.

● **Jak więc powstała firma Apple Computer?**

— Wraz ze Stevem rozdawaliśmy schematy komputera i terminala w Homebrew Computer Club, a nawet wręcz chodziliśmy do ludzi do domów, pomagając im budować i testować własne komputery. Wreszcie Steve powiedział: „Spójrz, ludzie są zainteresowani tym co zrobiliśmy. Dlaczego by nie zrobić gotowych płytek, opisać je, aby ludzie wiedzieli gdzie dołączyć jaką część i zacząć sprzedawać je w klubie?"

Klub liczył w tym czasie ok. 500 członków i sądziliśmy, że być może ok. 50 zdecyduje się to kupić. Obliczyliśmy sobie, że opracowanie takiej płytki kosztowałoby nas ok. 1000 dolarów, a wykonanie egzemplarza — ok. 20 dol. tak więc jeśli zdołalibyśmy sprzedać je 50 osobom po 40 dol., wówczas mielibyśmy swój tysiąc dolarów z powrotem. Wyglądało to więc na raczej kiepski interes, ale Steve zdołał mnie przekonać: „Tak czy siak, przynajmniej raz w życiu będziemy mieli własną firmę". Następnie Steve sprzedał swoją furgonetkę, a ja kalkulator Hewlett-Packarda i w ten sposób zdobyliśmy kapitał zakładowy.

Ledwie uruchomiliśmy interes, Steve zdobył duże zamówienie z lokalnego sklepu komputerowego na dostawę kompletnych, zmontowanych komputerów. Opiewało ono na bodajże 100 jednostek po 500 dol. każda, przy założeniu dalszej sprzedaży ich po 666 dol. To było wręcz niewiarygodne — zamówienie za 50 tys. dol.! Byliśmy więc przedsiębiorcami!

Najpierw jednak musieliśmy znaleźć 20 tys. dol. na części. Steve poszedł pertraktować z miejscowymi dostawcami i wypełniliśmy odpowiednie wnioski finansowe. Przyjrzeli się naszemu zamówieniu, podzwonili tu i ówdzie, zbadali nawet stan naszego zadłużenia w okolicznych sklepach, po czym ostatecznie dali nam części na 30-dniowy kredyt. Musieliśmy zgromadzić wszystkie potrzebne elementy oraz zmontować komputery w ciągu 10 dni — i czyniło nas to niewymownie szczęśliwymi. Dostarczyliśmy komputery w terminie, zapłaciliśmy dostawcom części. Po drodze musieliśmy pożyczyć jedynie 5000 dol. od naszego przyjaciela Alana Bauma i jego ojca.

● **Ile sztuk Apple I w końcu sprzedaliście?**

— Zrobiliśmy ich 200 i sprzedaliśmy wszystkie (poza może 25) w ciągu 9 lub 10 miesięcy.

● **Kiedy to było?**

— W 1976 roku. Po raz pierwszy zademonstrowaliśmy działający komputer pod koniec 1975 roku, a Steve wpadł na pomysł stworzenia firmy pod koniec 1976 roku. Oficjalnie spółkę założyliśmy w marcu. Mieliśmy z początku trzeciego partnera z udziałem w wysokości 10 proc., ale sprzedał swój udział za 800 dol., gdyż uważał, że nasza spółka nie prowadzi do niczego poza długami, a był on w niej jedynym partnerem z pieniędzmi.

● **Znamy co najmniej cztery lub pięć historyjek na temat pochodzenia nazwy Apple...**

— Wymyślił ją Steve Jobs, a że jest on typem człowieka raczej zamkniętego w sobie, do dzisiaj nie wiem co go na nią naprowadziło. Po prostu przyszedł z gotowym pomysłem. Od czasu do czasu dorabiał sobie pracą w sadach Oregonu przypuszczam, że może to mieć związek z jabłkami z tych sadów. Mogło się też zdarzyć, że po prostu spodobało mu się samo słowo. Tak czy siak obaj próbowaliśmy znaleźć lepszą nazwę, ale żaden z nas nie przypomina sobie, by przychodziło nam do głowy coś lepszego niż Apple.

● **Hewlett-Packard nie był zainteresowany prawami patentowymi do Apple I, mimo że zaprojektowałeś ten komputer podczas pracy dla tej firmy. Czy oferowałeś go im?**

— Tak. Było nas w laboratorium HP kilku zainteresowanych mikrokomputerami i proponowaliśmy ten temat kierownikowi laboratorium. Zorganizowaliśmy nawet spotkanie w tej sprawie, proponowaliśmy opracowanie dla HP małego komputera za 800 dol., który byłby w stanie realizować BASIC i być podłączony do domowego telewizora.

Facet ten był szefem zespołu opracowującego komputer stołowy HP 9830 i znał się na tych sprawach. Dobrze wiedział, dlaczego HP nie może robić takich rzeczy i miał rację. Hewlett-Packard po prostu nie mógł się „zniżyć" do poziomu hobbystycznych zabawek i nie miał ochoty ani potrzeby wkraczania na szybko ewoluujący, bardzo młody i niesprawdzony rynek mikrokomputerów. Nasz wniosek został więc nam zwrócony i dostałem formalne zwolnienie mojego opracowania od roszczeń prawnych pracodawcy.

Później zdarzyła się jeszcze zabawniejsza historia: kiedy zaczęliśmy już sprzedawać Apple I, nasz wydział kalkulatorów HP rozpoczął realizację małego projektu 8-bitowego procesora zwanego Capricorn. Miałem już za sobą większość drogi, którą oni od nowa, mozolnie przebywali, ale nie pozwolono mi pracować nad tym projektem.

Opr. WM



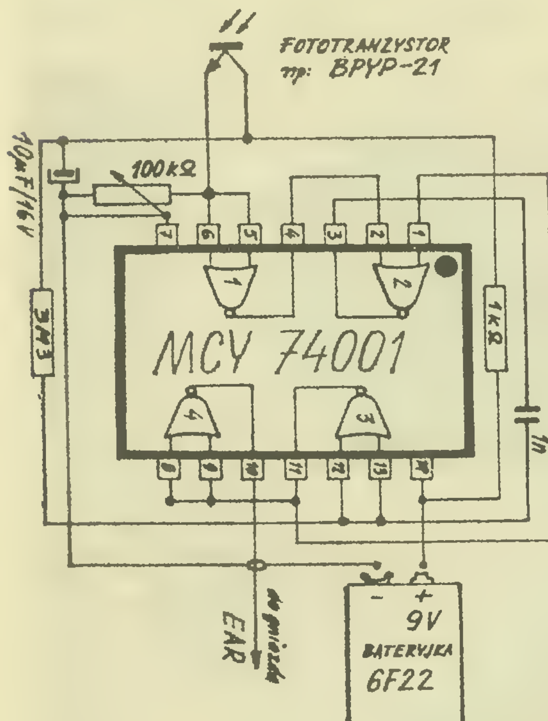
PIÓRO ŚWIETLNE

Pióro świetlne jest jednym z najbardziej atrakcyjnych urządzeń peryferyjnych, dostępnych amatorowi zabawy i pracy z mikrokomputerem. Wbrew pozorom, jego wykonanie nie wymaga specjalnych umiejętności, czy też teoretycznego przygotowania z zakresu elektroniki.

Ogólna zasada działania pióra jest bardzo prosta. Impulsy świetlne wysyłane przez komputer na ekran są za każdym razem odpowiednio adresowane, a więc zawsze „wie" on, który element ekranu jest w danym momencie pobudzony do świecenia. Jeżeli równocześnie uzyska sygnał z czujnika przyłożonego do ekranu, wówczas punkt ten jest jednoznacznie określony. Reszta to sprawa odpowiedniego programu.

Schemat elektryczny naszego urządzenia przedstawiamy na rys. 1. Jako czujnika użyto w tej konstrukcji fototranzystora (np. BPYP-21). Ze względu na bezpieczeństwo (komputera a nie użytkownika) proponujemy zastosowanie osobnego źródła zasilania w postaci bateryjki 6 F 22 o napięciu 9 V. Podłączanie się bezpośrednio do szyny kontaktowej, przy nieostrożnej obsłudze, mogłoby spowodować nader przykre konsekwencje.

Możemy zdecydować się na jedną z dwóch technologii: montaż układu elektronicznego na



Rys. 1. Schemat elektryczny pióra świetlnego.

płytce drukowanej lub „upakowanie" wszystkiego w samym piórze.

■

W tym drugim przypadku konieczne jest zastosowanie podstawki do układu scalonego, do której przylutujemy poszczególne elementy. Rozpoczniemy od połączenia ze sobą nóżek: 8, 9, 11 i 1 oraz parami 6 i 7 a także 2 i 4. Wszystkie masy i ekrany oraz potencjometr montażowy o pojemności 100 kiloomów, lutujemy do nóżki 7. Pozostałe elementy montujemy wg rysunku 2. Na odpowiednio długim przewodzie umieszczamy wtyczkę. W przypadku ZX Spectrum będzie to tzw. „mały Jack", włączany do gniazda EAR. Baterię możemy przymocować bezpośrednio do pióra, jednakże ze względu na jej wymiary nie jest to najlepsze rozwiązanie.

Nie da się ukryć, że zlutowanie opisanego układu będzie wymagało nieco cierpliwości i staranności. Nóżki podstawki umieszczone są blisko siebie, a całość — z konieczności — musimy zmieścić w niewielkiej przestrzeni (np. wewnątrz grubego pisaka).

Jeśli będziemy już w 100% pewni sprawności działania naszego układu, możemy zabezpieczyć się przed wszelkimi niespodziankami, pokrywając wszystkie połączenia klejem Epidian. Uniemożliwi to powstawanie przypadkowych spięć.

■

Lepszym rozwiązaniem, choć wymagającym nieco więcej zachodu, jest zmontowanie układu na płytce. Rys. 3 przedstawia płytkę od strony ścieżek (tzn. od strony folii miedzianej). Rozpoczynamy od wykonania w niej otworów wiertłem o średnicy 0,8—1 mm. Następnie czyścimy i odtłuszczamy bardzo starannie naszą płytkę i za pomocą farby (np. nitro) nanosimy na nią rysunek ścieżek.

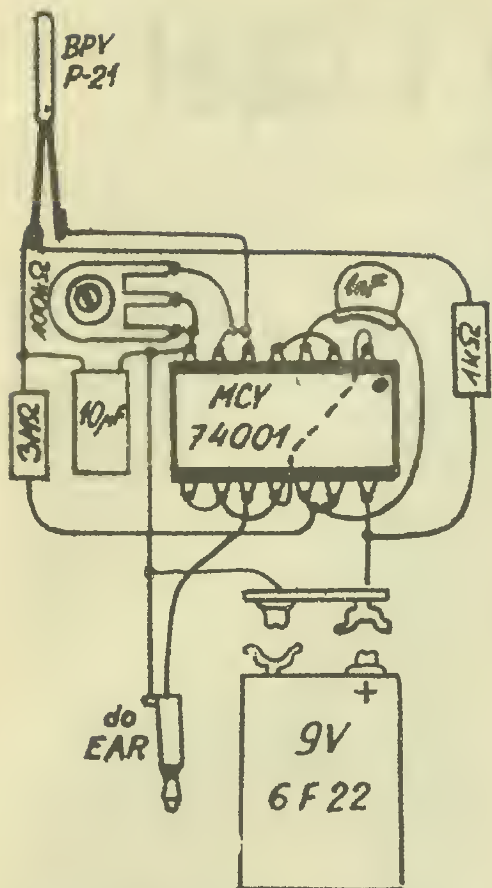
Po całkowitym wyschnięciu farby — to bardzo ważne, gdyż w przeciwnym przypadku możemy zniszczyć efekt naszej dotychczasowej pracy — przystępujemy do wytrawiania. Polega ono na usunięciu nie zamalowanych powierzchni miedzi poprzez reakcję z silnym utleniaczem. Najlepiej nadaje się do tego celu chlorek żelazowy. W przypadku trudności z jego zdobyciem, można zastosować roztwór kwasu azotowego, lecz wówczas należy postępować o wiele ostrożniej, tak ze względu na możliwość zniszczenia płytki, jak i własne bezpieczeństwo. Stężenie najlepiej jest dobrać doświadczalnie, zwiększając je powoli, aż do zaobserwowania wyraźnych, choć nie gwałtownych objawów reakcji (bąbelki na powierzchni miedzi).

Po zakończeniu wytrawiania, płytkę należy bardzo dokładnie wypłukać, usunąć farbę, przetrzeć ścieżki drobnym papierem ściernym i możemy przystąpić do wlutowania poszczególnych elementów. Ich rozmieszczenie przedstawiono na rys. 4.

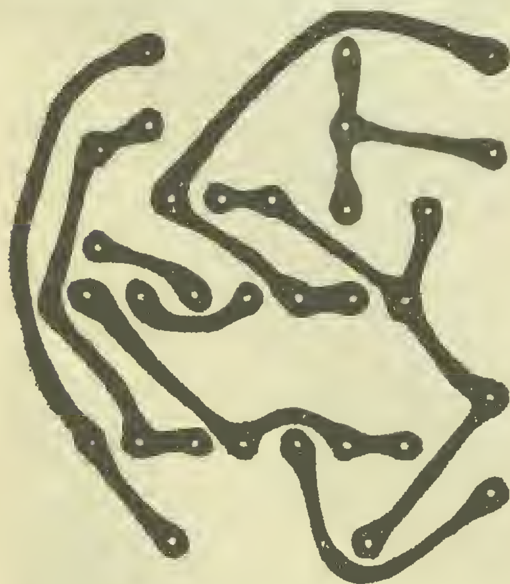
Oznaczenia:
R_1 — 1kΩ
R_2 — 3,3 MΩ
C_1 — 10 μF (elektrolityczny)
C_2 — 1 nF
P — 100kΩ (montażowy)

Płytkę najlepiej umieścić w jednej obudowie wraz z baterią. Jeśli wykonamy nieco dłuższą płytkę, wówczas baterię będziemy mogli przymocować bezpośrednio do niej. Szczegółowe rozwiązanie pozostawiam czytelnikowi.

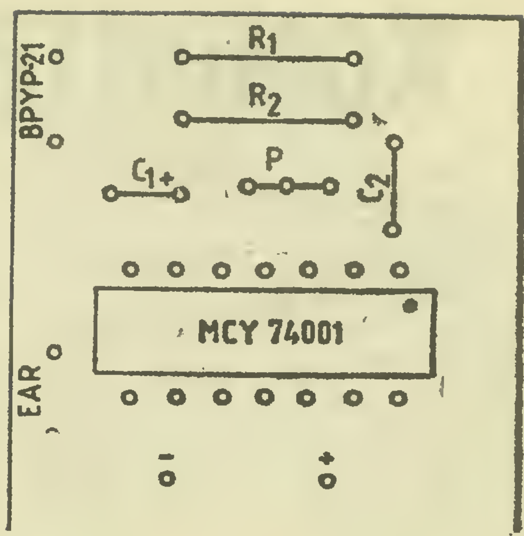
Warto zwrócić uwagę na sposób umocowania fototranzystora (rys. 5), jego koniec powinien znajdować się na równi z krawędzią osłony.



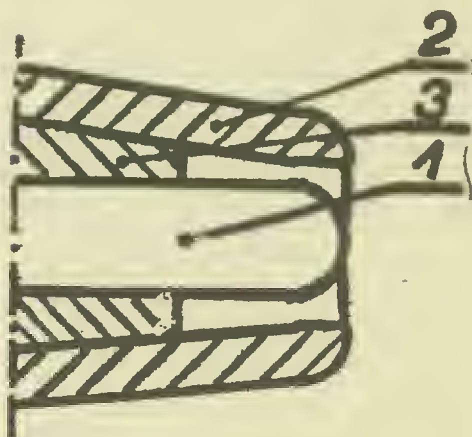
Rys. 2. Rysunek poglądowy, przedstawiający sposób montażu bezpośrednio do podstawki układu scalonego.



Rys. 3. Płytka drukowana — widok od strony miedzi (podz. 2:1).



Rys. 4. Płytka drukowana — widok od strony elementów (podz. 2:1).



Rys. 5. Sposob zamocowania fototranzystora. 1 — fototranzystor, 2 — obudowa, 3 — podkładka gumowa.

Powyższy opis wykonano na podstawie dokumentacji udostępnionej przez Agencję Mikrokomputerową AMICO, Sosnowiec P-157, tel. 699-649. Agencja dysponuje również programami użytkowymi, wykorzystującymi pióro świetlne.

ROMAN POZNAŃSKI

KOMPUTER I GĘSI

Niechaj narodowie wżdy postronni znają, iż Polacy nie gęsi... powtarzaliśmy wielokrotnie dyskutując nad stanowiskiem Prezydium ZG Polskiego Towarzystwa Informatycznego w sprawie wprowadzenia elementów informatyki do szkolnictwa średniego i podstawowego.

Część naszych kolegów i Klaksona ze „Szpilek" zbulwersował bowiem fragment stwierdzający: „uważamy za niezbędne przestrzec władze oświatowe PRL przed fatalnymi skutkami wprowadzenia do szkół (...) urządzeń komputerowych nie akceptujących polskiego alfabetu i wymagających obcojęzycznych instrukcji. Wywoływanie przeświadczenia, że nowoczesność wymaga lekceważenia języka ojczystego jest pedagogicznym przestępstwem".

PTI nawiązywało tu do dobrze znanych faktów, towarzyszących wprowadzeniu do naszych szkół sprzętu Sinclaira. Spora grupa czytających odebrała jednak cytowany fragment jako wyraz tęsknoty do zaściankowości, odcięcia się od świata. Argumentowano, że angielski jest międzynarodowym językiem informatyki i że tłumaczenie BASIC-u na polski to nieporozumienie. Istotnie — ale takich zamiarów nikt nie ma.

Dziś jednak, na naszych oczach, komputery przestają być narzędziem pracy nielicznych użytkowników, obsługiwanym przez wysoko wtajemniczonych kapłanów, nie dopuszczających nikogo w pobliże bożka. Zaczynają być fragmentem środowiska kulturowego każdego człowieka. Nie można już więc wymagać, by człowiek ten o swym narzędziu i za jego pośrednictwem wypowiadał się w mało przypominającym ojczysty język slangu. Nikt przecież nie uważa, że ze względu na miejsce wynalezienia telefonu czy gramofonu należy przekazywać za ich pomocą tylko anglojęzyczne rozmowy i piosenki ani też nie uznałby za normalne dostarczanie wraz z tym urządzeniem obcojęzycznej instrukcji lub rachunku (choć niektóre polskie płyty mają obcojęzyczny tytuł, to jednak nikt tego nie wiąże z naturą gramofonu...).

Ze swym użytkownikiem polski komputer powinien się porozumiewać po polsku. Na ekranie polskiego nigdy, nigdzie nie eksportowanego popularnego urządzenia komunikat „out of memory" lub „syntax error" zamiast „brak pamięci" lub „błąd składni" zakrawa na taki sam dowcip co etykieta „Warka export full light beer". Przygotowane na tenże komputer polskie i przewidziane do obsługi przez Polaków programy, zaopatrzone w odautorskie komentarze w niby-angielskim budzą już tylko politowanie. Przed kim ci ludzie — a jest to zjawisko masowe — chcą się popisać? Dla kogo wygodniejsze lub bardziej naukowe jest odczytanie z ekranu „biorythmes" lub udzielanie odpowiedzi na pytanie „your name"?

Z innych nieco przesłanek wychodzi postulat polonizacji LOGO. Doświadczenia pedagogiczne uczą, że entuzjazm komputerowy wcale nie jest wśród młodzieży tak powszechny, jak to się nam na pozór wydaje. Ścisk wokół stoisk z komputerami i w szkolnych pracowniach jest dziełem około 1/4 młodzieży (głównie chłopców). Reszta pozostaje wroga lub obojętna: skłonna bez entuzjazmu, ale z zainteresowaniem nauczyć się czegoś, gdyby nauka ta nie była zbyt męcząca i zniechęcająca. Obce słowa tej milczącej większości nie przyciągają, lecz odtrącają. Ten komu zależy więc na masowym odbiorze tej nauki, musi ją uczynić przystępną i zrozumiałą a za to przynoszącą szybko odczuwalne efekty mogące być zachętą do dalszej pracy.

Każdy kij ma dwa końce — w polonizacji terminologii też można przesadzić. Zastępując powszechnie uznane terminy informatyczne („byte" po angielsku też nic innego nie znaczy) przez skrótowce polskie, na ogół niemożliwe do wymówienia (brak samogłosek). T[illegible] zastąpiono wprowadzając polskie mnemoniki rozkazów produkowanego w [illegible] polskiego mikroprocesora, działaniem przypominającego Intel 8080. Czy zabieg ten miał ukryć związki z pierwowzorem?



Równie zabawna [illegible] zastąpienia rozkazu „end" skrótem „kne", czy też zastąpienia komendy [illegible] (erase all) przez [illegible] (wyrzuć [illegible]) tego [illegible] w determinację i wyrzucić wszystko, a zwłaszcza... komputer za [illegible]

Wielu zwolenników anglojęzyczny informatycznej [illegible], że terminologia [illegible] młodzież do nauki języków obcych. Wszelkie złudzenia pryskają w konfrontacji z praktyką tychże lingwistycznych umiejętności. Gdy dziesiąty raz usłyszałem od ucznia, że po [illegible] (czytaj: po wykonaniu instrukcji GO TO) program nie daje się zbrekować (czyli nie działa przycisk BREAK), sam [illegible] zaproponować delikwentowi, by wypokował (od POKE) sobie na brajcie (od: BRIGHT) [illegible] nie wystarcza do sklirowania (CLEAR) zmiennych. Efekt dydaktyczny był natychmiastowy. Oczy zapalały mu się i gasły jak po flaszy (FLASH), przez chwilę wyglądał jak inwers (INVERSE), ale w końcu mu chyba coś w tej jego ramie (RAM, Random Access Memory) pozostało. Mnie też.

WŁADYSŁAW MAJEWSKI

SZANSA

Zapewne nie raz zdarzyło się Wam prosić rodziców o pieniążki na kupno różnych, niezbędnych rzeczy. Uzyskiwane odpowiedzi rzadko kiedy mogą nas usatysfakcjonować, sprowadzają się bowiem zazwyczaj do mniej lub bardziej eleganckiej odmowy. Zanim wreszcie wyprosimy potrzebną sumę, niepowtarzalna okazja ucieka, a w pamięci pozostaje jedynie obiecanka: „jak wygram w TOTO-LOTKA to ci dam." Ciężko jest znaleźć na to jakąś skuteczną radę, ale być może przy pomocy ulubieńca całej rodziny — komputera, uda się nam odnieść pewne umiarkowane sukcesy.

Zakradnijmy się więc do owej cudownej skrzynki i rozpocznijmy budowę programu, który może przekona Tatę o konieczności wpłacenia nam żądanej sumy bez czekania na wielką (a choćby i mniejszą) wygraną w TOTO-LOTKA.

Tym razem wykorzystamy zdolność komputera do zapamiętania danych. Zaczynamy od naszych typów. Grupa rozkazów 1Ø — 4Ø zmusza komputer do żądania od nas podania naszych numerów. Spotykamy tu instrukcję FOR i TO. Wykonując ją komputer realizuje zadaną czynność określoną ilość razy. Rozkaz 1Ø możemy więc przetłumaczyć jako: FOR(DLA) K=1 TO(DO) 6. W późniejszych rozkazach spotkamy się też z instrukcją STEP, która określa, co ile ma komputer zwiększać owo K (albo inną zmienną). Np. FOR K = 1 TO 22 STEP 3 nasza maszyna policzy od 1 do 22 przeskakując co 3, a więc: 1, 4, 7 itd.

Wykonując rozkaz 2Ø komputer nie przestanie nas męczyć, dopóki nie podamy mu owych 6 liczb N. Będą to nasze typy w losowaniu. Chcemy oczywiście, aby zostały one wyświetlone na ekranie. Służy temu rozkaz 3Ø, gwiazdka w nim jest konieczna, inaczej bowiem podane przez nas numery zleją się razem i nie będziemy wiedzieli, co właściwie jest naszym typowaniem.

Przechodzimy do losowania. Sprawa jest o tyle skomplikowana, że musimy zmusić komputer do wybrania 6 różnych liczb. Ale i na to jest rada. Tu właśnie wykorzystamy zdolność komputera do zapamiętywania, a także porównywania liczb. Wszystko to zawieramy w rozkazach 60 — 11Ø. Właściwe losowanie wykonuje rozkaz 7Ø, numery 8Ø i 9Ø zabezpieczają losowanie przed powtorzeniem takich samych numerów, rozkazy 6Ø, 1ØØ, 11Ø powodują losowanie 7 numerów (ostatni potraktujmy jako dodatkowy), oraz uruchamiają tę część programu. Najciekawszy, a jeszcze nam nieznany jest rozkaz 1. Owo DIM oznacza, że nasza maszyna musi zapamiętać 7 liczb, określanych w tym wypadku jako Y. Gdyby chodziło o pamiętanie wyrazów, albo ogólnie liter, polecenie to wyglądałoby na przykład tak: DIM Y$(3,5). Powodowałoby to zapamiętanie 3 słów (lub grup liter), każde złożone z 5 liter.

Musimy oczywiście zobaczyć, jakie liczby wylosował nasz przyjaciel komputer. Służą temu rozkazy 12Ø — 14Ø. Teraz okaże się, jak bardzo przydatne jest numerowanie rozkazów co 10, korzystając bowiem z możliwości uzupełnień, wprowadzimy napisy objaśniające. Są to komendy 8 PRINT „TWOJE TYPY:" i 115 PRINT „WYLOSOWANO:". Powoli zbliżamy się do finału. Aby nie trudzić się zbytnio, zaprogramujemy komputer tak, aby powtarzał grę określoną przez nas ilość razy. Musimy więc policzyć ilość gier, określimy je jako T. Na początku jest oczywiście T=0 (rozkaz 3), następnie co gra to wartość T rośnie o jeden. (Rozkaz nr 5). W komendzie 15Ø wyznaczyliśmy liczbę gier na 5, możecie oczywiście to zmienić, gdy T=5 program kieruje się do ostatniego rozkazu, jeżeli jeszcze nie, powtarza program od nr 5 (rozkaz 17Ø). Ilość gier (T) wyświetlimy sobie w prawym, dolnym rogu ekranu, a mnożąc ją przez opłatę jednej gry, otrzymamy sumę wydatków. Jest to rozkaz 16Ø. T, czyli ilość gier w naszym przykładzie mnożymy przez 40 zł, czyli połowę stawki, bo nasz program wykonuje dla uproszczenia tylko jedno, a nie dwa losowania.

Wszystko gotowe!

Wystarczy teraz, że uruchomimy program i poprosimy rodziców o wytypowanie liczb. Na ogół suma w prawym, dolnym rogu rośnie i rośnie, a trafnych „trojek" i ew. większych wygranych nie ma i nie ma. Wtedy skromnie prosimy rodziców o przekazanie tej sumki na nasze konto i pędzimy zakupić ową niepowtarzalna okazję.

No, chyba że nie mamy szczęścia (a rodzice je mają) i stwierdzamy jakieś wygrane. I tu właśnie przejawia się wyższość TOTO-LOTKA, bo on wypłaca wygrane, a my nie.

```
1 DIM Y (7)
3 LET T = Ø
5 LET T = T + 1
8 PRINT "TWOJE TYPY";
1Ø FOR K = 1 TO 6
2Ø INPUT N (K)
3Ø PRINT N (K); "*";
4Ø NEXT K
6Ø FOR X = 1 TO 7
7Ø LET Y (X) = INT (RND * 49) + 1
8Ø FOR Y = X — 1 TO 1 STEP — 1
9Ø IF Y (X) = Y (Y) THEN GOTO 7Ø
1ØØ NEXT Y
11Ø NEXT X
115 PRINT „WYLOSOWANO:"
12Ø FOR X = 1 TO 7
13Ø PRINT Y (X); "*";
14Ø NEXT X
15Ø IF T = 5 THEN GOTO 18Ø
16Ø PRINT AT 21, 11; "GIER: "; T ; AT 21,
19; „WYDATKI: "; T * 4Ø
17Ø GOTO 5
18Ø PRINT „DZIEKUJE."
```

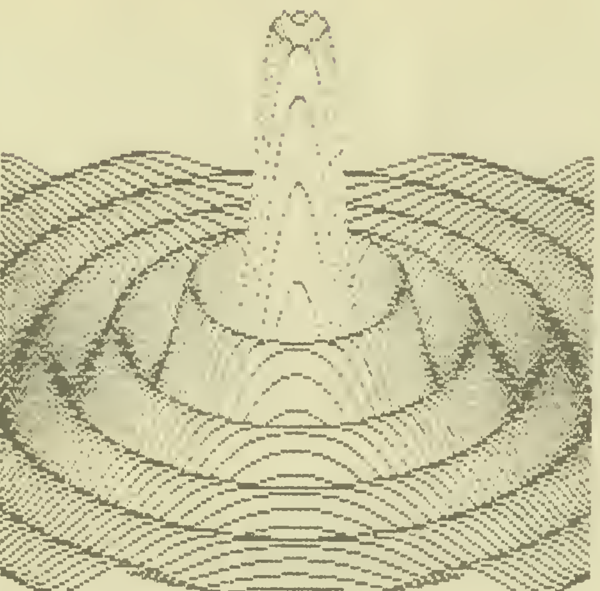
MAREK

W następnym numerze m.in.:

- micro-PROLOG cd.
- Olimpiada informatyczna
- Odpowiadamy czytelnikom
- FLOWCHART
- Bez barier — kopiowanie programów
- Komputery w Chinach
- „Tylko dla przedszkolaków"

i inne stałe rubryki

GRAFIKA



```
** Econet ** Station   [illegible] **

..
  20MODE 0:VDU19,0,4;0;
  50XM=640.1
  60ZM=800
  70YM=700
  80TILT=20
  90ST=SIN(RAD(TILT)):CT=COS([illegible]))
 100VDU29,640;300;
 110FORX=-XM TO 0 STEP 4
 120FORY=-ZM TO ZM STEP 16
 130SCREENY=Y*ST+CT*YM*FNsin(X,Y)
 140 IFY=-ZM THEN MINY =SCREENY:M[illegible]CREENY
 150IF SCREENY>MAXY THEN MAXY=SC[illegible]:PROCPLOT
 160IF SCREENY<MINY THEN MINY=[illegible]:PROCPLOT
 170NEXTY
 180NEXTX
 190VDU5:MOVE0,1050
 200*SAVE"RIFFLE" 3000 9000 [illegible] 7000 [illegible] 000
 202 END
 210DEFPROCPLOT
 220PLOT69,X,SCREENY:PLOT69,-X,[illegible]NY
 230ENDPROC
 240:
 250DEFFNsin(X,Y)
 260KR=.02*[illegible]R(X*X+Y*Y)
 270=SIN(KR)[illegible]/KR
```

Taką trójwymiarową grafikę wraz z programem (BASIC — wersja BBC) można było zobaczyć podczas seminarium brytyjskiej firmy 3SL poświęconemu komputerowemu wspomaganiu nauczania. Odbywało się ono w ostatnich dniach lipca br. w Instytucie Maszyn Matematycznych w Warszawie, a towarzyszył mu pokaz systemu mikrokomputerowego firmy BBC. Można było także indywidualnie popracować na sprzęcie, zapoznając się z możliwościami systemu oraz interesującymi programami edukacyjnymi.